

Rapporti tecnici

INGV

TSDSystem:
un database multidisciplinare per
la gestione di serie temporali

304



Istituto Nazionale di
Geofisica e Vulcanologia

Direttore Responsabile

Stefano Gresta

Editorial Board

Luigi Cucci - Editor in Chief (INGV - RM1)

Andrea Tertulliani (INGV - RM1)

Nicola Pagliuca (INGV - RM1)

Umberto Sciacca (INGV - RM2)

Alessandro Settimi (INGV - RM2)

Aldo Winkler (INGV - RM2)

Salvatore Stramondo (INGV - CNT)

Milena Moretti (INGV - CNT)

Gaetano Zonno (INGV - MI)

Viviana Castelli (INGV - BO)

Antonio Guarnieri (INGV - BO)

Mario Castellano (INGV - NA)

Mauro Di Vito (INGV - NA)

Raffaele Azzaro (INGV - CT)

Rosa Anna Corsaro (INGV - CT)

Mario Mattia (INGV - CT)

Marcello Liotta (INGV - PA)

Segreteria di Redazione

Francesca Di Stefano - Referente

Rossella Celi

Barbara Angioni

Tel. +39 06 51860068

Fax +39 06 36915617

redazionecen@ingv.it

Rapporti tecnici

INGV

TSDSYSTEM: UN DATABASE MULTIDISCIPLINARE PER LA GESTIONE DI SERIE TEMPORALI

Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, Michele Prestifilippo

INGV (Istituto Nazionale di Geofisica e Vulcanologia, Sezione di Catania - Osservatorio Etneo)

304

Indice

| | |
|---|----|
| Introduzione | 7 |
| 1. Struttura concettuale del sistema TSDSystem | 7 |
| 2. Modello Entità-Relazione di TSDSystem | 8 |
| 3. I <i>Loader</i> | 10 |
| 3.1 Stato di funzionamento dei <i>loader</i> : il sistema <i>heartbeat</i> | 12 |
| 3.2 Interfaccia di configurazione del <i>loader</i> | 17 |
| 4. Applicativi per la visualizzazione e l'interrogazione dei dati di <i>TSDSystem</i> | 20 |
| 4.1 L'applicativo <i>TSViewer</i> | 21 |
| 4.2 Utilizzo di <i>TSViewer</i> per il monitoraggio | 24 |
| 4.3 Interfaccia web per la visualizzazione delle serie temporali | 27 |
| 5. Conclusioni | 31 |
| Ringraziamenti | 31 |
| Bibliografia | 31 |

Introduzione

Le grandezze acquisite da una rete di sensori costituiscono una fonte dati eterogenea che va organizzata opportunamente al fine di poter essere impiegata nell'ambito della sorveglianza, del monitoraggio e della ricerca in ambito geofisico. Con il termine *serie temporale* ci si riferisce ad un insieme di misure relative ad un determinato fenomeno, acquisite sequenzialmente nel tempo. Quando gli intervalli temporali sono equispaziati si parla di *periodo* o di *frequenza* di campionamento.

Il presente report tecnico descrive nel dettaglio una possibile metodologia di archiviazione e gestione delle serie temporali mediante l'impiego di una specifica struttura dati.

Il sistema prodotto, di seguito *TSDSystem* (*Time Series Database System*), è progettato in modo da acquisire serie temporali da differenti sorgenti dati e di standardizzarle all'interno di un database relazionale. L'operazione di standardizzazione fornisce la possibilità di eseguire operazioni come la richiesta e/o la visualizzazione. Tali operazioni possono essere eseguite contemporaneamente su qualsiasi serie temporale presente all'interno del database, permettendo di utilizzare un asse temporale comune.

L'architettura proposta segue un paradigma composto da più livelli, detti *layers*, ognuno specializzato nell'eseguire particolari operazioni per la riorganizzazione e l'archiviazione di dati provenienti da fonti differenti quali file ASCII, file Excel, database ODBC (*Open DataBase Connectivity*), o file accessibili dal Web.

1. Struttura concettuale del sistema TSDSystem

La struttura concettuale del sistema *TSDSystem* è costituita da 5 layers (Figura 1). Il primo layer, chiamato *Data Source*, è relativo alle diverse sorgenti dati disponibili. Al momento della stesura del report le sorgenti dati considerate sono file ASCII, origini dati ODBC e dati in formato ASCII accessibili dal Web. I layers intermedi che costituiscono *TSDSystem* sono: *Loader*, costituito da tutti i moduli di importazione verso i database; *Database*, ossia l'organizzazione logica e fisica dei dati mediante l'utilizzo di database relazionali; *Business Logic*, composto da tutti i moduli che si occupano di organizzare i dati richiesti dal layer *Applications*. In particolare, quest'ultimo è costituito da tutte le applicazioni utente che richiedono i dati archiviati all'interno del sistema come, ad esempio, software di consultazione e visualizzazione. *TSDSystem* nasce come estensione del database relativo all'archiviazione delle informazioni riguardanti le reti strumentali in forza all'Istituto Nazionale di Geofisica e Vulcanologia - Osservatorio Etneo, Sezione di Catania. Tale database, come descritto in [Aliotta et al., 2010], è progettato al fine di archiviare tutte le informazioni relative ai sensori, alla loro ubicazione e alle loro caratteristiche tecniche.

L'entità utilizzata per descrivere un sensore (entità *sensor*), prevede come attributo il numero di componenti dati da acquisire. A titolo di esempio, è disponibile l'entità *velocimeter* definito come sensore a 3 componenti (una verticale e due orizzontali). Ad una entità *sensor* è possibile associare una o più entità *channel*, ciascuna avente come attributi: l'identificativo del canale, una descrizione, la frequenza di campionamento, il tipo di acquisizione etc. Ad esempio, ad un sensore sismico velocimetrico a 3 componenti vengono assegnati 3 canali di acquisizione, ognuno con le caratteristiche specificate dagli attributi della relativa entità. Quanto finora esposto è esteso a tutte le tipologie di sensori caricati all'interno della banca dati. Nell'evoluzione proposta, l'entità canale è legata ai dati acquisiti dallo stesso. In questo modello, un canale è logicamente legato ai dati acquisiti ed ai relativi dati elaborati. Consideriamo una grandezza x acquisita con una frequenza di campionamento f_c . La sequenza delle misure che costituisce la serie temporale archiviata è logicamente legata al canale impiegato per l'acquisizione. Oltre al dato grezzo acquisito (dato *raw*), allo stesso canale possono essere logicamente connesse anche serie temporali ricavate da processi di elaborazione sul dato grezzo. In questo modello, un canale di acquisizione sarà messo in relazione ad n possibili serie temporali. Un esempio può essere l'acquisizione della velocità del suolo rispetto alla componente verticale mediante un velocimetro e successive elaborazioni, quali RMS (*Root Mean Square*) del segnale sismico ricavata mediante una finestra mobile di lunghezza T , oppure la stima delle variazioni dei *trend* mediante algoritmi di segmentazione delle serie temporali [Montalto et al., 2011]. Analogamente è stato implementato il caso in cui più canali di acquisizione sono legati ad un'unica serie temporale. Ad esempio, si può considerare la serie temporale della localizzazione della sorgente di tremore vulcanico eseguita impiegando il segnale sismico acquisito sulla componente verticale di più stazioni. In questo caso la variazione delle coordinate ipocentrali [Cannata et al., 2013] nel tempo è data da 3 serie temporali: latitudine, longitudine e profondità; ognuna delle quali associata ai canali delle stazioni impiegate nell'algoritmo di localizzazione.

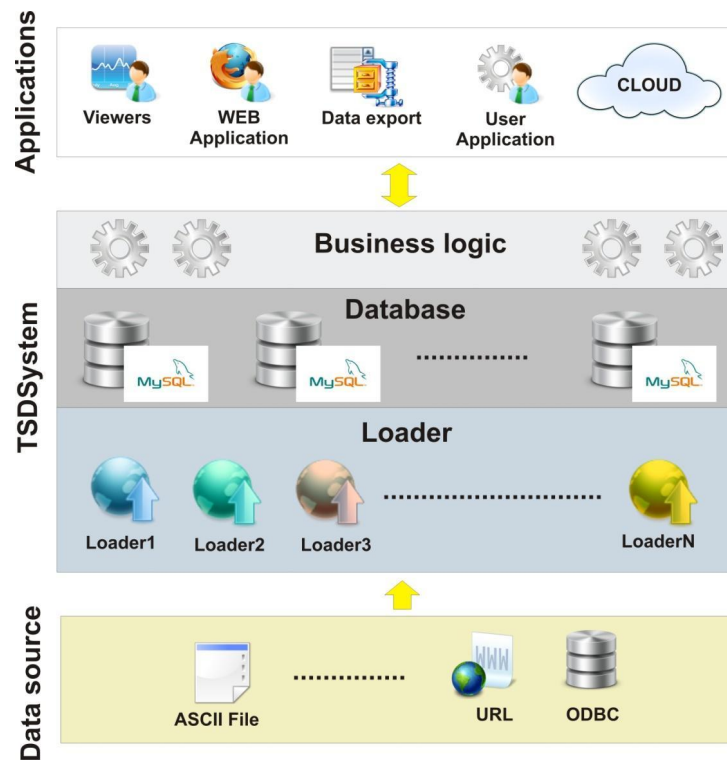


Figura 1. Schema concettuale del framework *TSDSystem*.

2. Modello Entità-Relazione di TSDSystem

Le relazioni descritte nel paragrafo precedente sono riportate in Figura 2a, mediante uno schema entità-relazione. L'associazione tra canale e relative serie temporali prende il nome di relazione *N-N*: un canale è associato a più serie temporali, una serie temporale è associata a più canali. Tale associazione è modellata introducendo la relazione *data_channel_mapping*; dato un canale associamo i dati modellati con l'entità *Data_Table*. Per ogni dato (serie temporale associata) la relazione fornisce informazioni specificate tramite appositi attributi come l'*host (server)* in cui i dati sono archiviati, e informazioni sul tipo di dati modellati mediante l'entità *Data_Type*. L'attributo *dataSplit* è utile per ciò che concerne l'organizzazione dei dati, in quanto codifica le informazioni relative al partizionamento dei dati stessi. Dopo la definizione del modello entità relazione occorre passare alla progettazione logica della base di dati, ovvero definire lo schema logico dipendente dal particolare sistema scelto per l'implementazione del database. In questa fase ogni entità e relazione viene rappresentata con una tabella le cui colonne sono gli attributi stessi. Lo schema logico va integrato con specifiche relative ai parametri fisici impiegati per la memorizzazione delle informazioni, come l'organizzazione dei file dove fisicamente risiederanno le tabelle e degli indici. In quest'ultima fase viene definito il modello fisico dei dati ed è strettamente legata alla scelta dell'RDBMS (Relational Database Management System).

Al fine di non introdurre ulteriori notazioni, gli autori del presente report preferiscono descrivere le scelte progettuali relative all'implementazione logica e fisica facendo riferimento a degli schemi derivati dal modello E-R di Figura 2a. Sebbene si stia impiegando un 'abuso' nella notazione introdotta, gli schemi riportati in Figura 2b e 2c esplicano le scelte progettuali effettuate dai progettisti del sistema.

Nella sua forma più semplice l'entità *Data_Table* può essere implementata fisicamente mediante l'impiego di una tabella i cui campi principali sono il valore (*attributo cValue*) del dato all'istante *t* (attributo *bTime*). Questa impostazione soffre di problemi intrinseci legati alla modalità di memorizzazione e di interrogazione. Il primo problema è relativo alla dimensione ed alla velocità di crescita della tabella stessa; più lungo è il periodo da archiviare, maggiore sarà lo spazio su disco richiesto per la memorizzazione dei dati; più alta è la frequenza di campionamento dei dati, più veloce sarà la crescita, in termini di spazio, della tabella stessa. Al fine di velocizzare le ricerche all'interno del database e, considerata la dimensione a cui possono arrivare le tabelle (anche ordini di parecchi Gigabyte) è necessario operare delle scelte progettuali.

Una prima idea potrebbe essere quella di indicizzare il campo relativo alla data. Negli RDBMS vengono introdotte delle strutture dati chiamate "indici", al fine di velocizzare le ricerche inviate mediante

query al database, all'interno dei dati. Sebbene l'impiego degli indici aiuti a ridurre l'insieme dei dati da leggere per eseguire una ricerca, la loro introduzione rende più lente le operazioni di inserimento e di modifica e comporta anche un aumento, in termine di memoria di massa, dello spazio richiesto per l'archiviazione dati. La seconda ipotesi di progetto prevede di dividere le tabelle per tipologia di dato e di ubicare le stesse su differenti macchine server (*host*).

Nello schema di Figura 2b (con abuso di notazione), viene modificato lo schema del modello E-R di Figura 2a specificando l'ubicazione della tabella relativa al *Data_Table* su un *host* differente. La terza specifica di progetto riguarda il partizionamento delle tabelle dati. Questa scelta progettuale è cruciale per ciò che riguarda le performance dell'intero sistema, in quanto razionalizza lo spazio di allocazione dei dati. Scegliendo di partizionare la *data_table* si ottengono numerosi vantaggi riguardanti la gestione del dato: è possibile, ad esempio, definire i periodi di partizionamento; in questo caso, per dati acquisiti a bassa frequenza (*low rate*) si possono scegliere delle tabelle con un periodo temporale più lungo rispetto a dati acquisiti ad alta frequenza (*high rate*). Altro vantaggio risiede nel fatto che i periodi di acquisizione così partizionati possono essere più facilmente posti in backup e/o spostati su altri server.

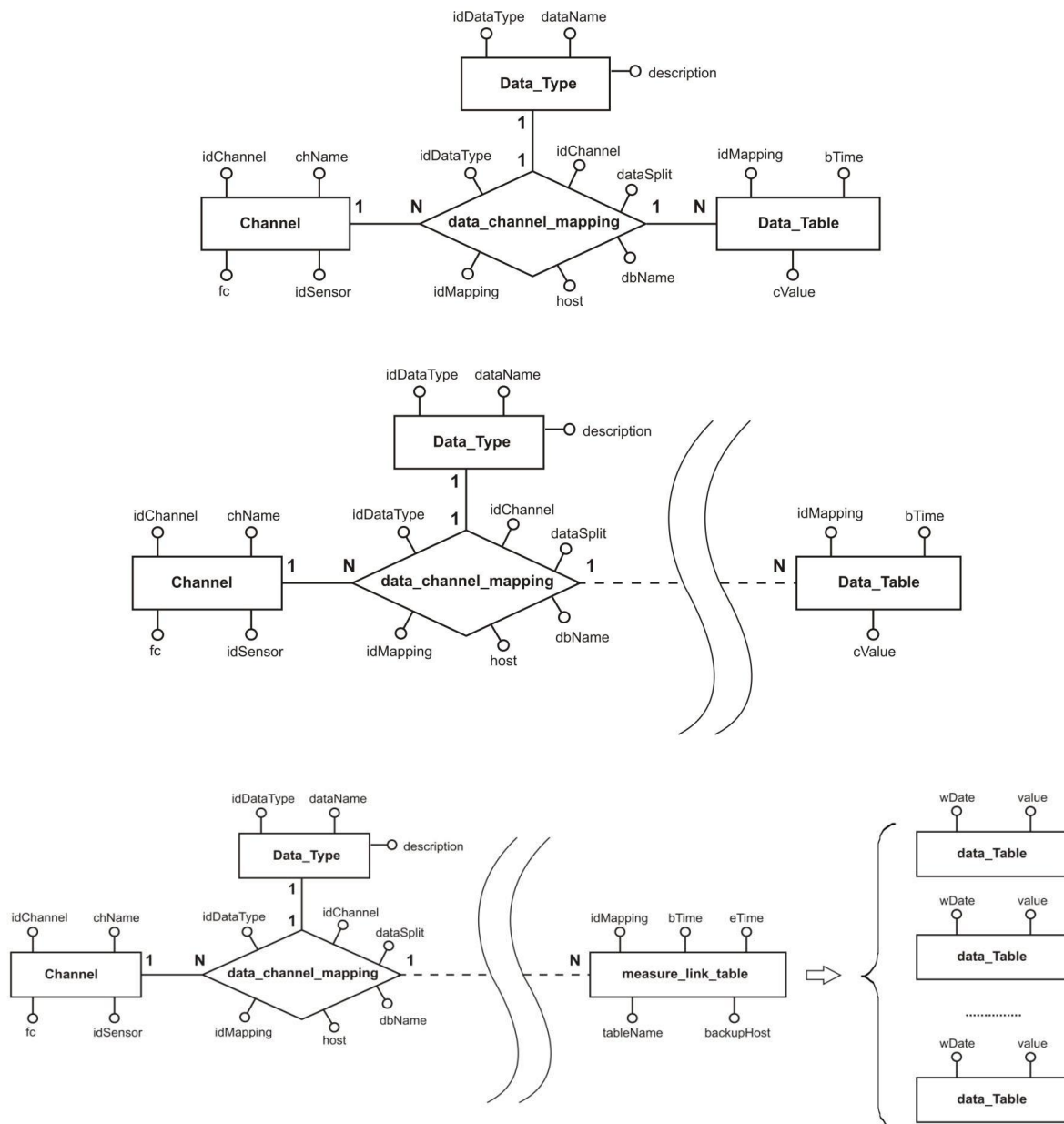


Figura 2. a) Modello E-R di TSDSystem; b) Modello E-R in cui la *Data_Table* risiede su un host remoto; c) partizionamento della *data_Table*.

Il partizionamento si rivela utile anche per mantenere un buon bilanciamento del carico di richieste provenienti dal layer *applications*. In questo caso, in base ai periodi richiesti, la ricerca di un determinato periodo temporale riguarderà solo una parte delle tabelle del database; nel caso in cui i database siano distribuiti su più *host*, il bilanciamento delle richieste sarà ancora più efficiente. In quest'ottica si può considerare, ad esempio, di rendere disponibili i dati degli ultimi mesi su alcuni *host* e di spostare su altri *host* i dati "storici"; in questo modo tutte le richieste riferite a periodi antecedenti saranno processate separatamente da quelle riferite al periodo attuale.

Il partizionamento delle tabelle può essere eseguito direttamente dall'RDBMS utilizzato, se esso implementa la funzionalità del *partitioning*, o gestito all'interno della politica di inserimento dei moduli progettati per il caricamento dei dati. Nel primo caso i progettisti si devono "affidare" al meccanismo di *partitioning* messo a disposizione dall'RDBMS utilizzato; nel secondo caso è compito dei progettisti definire ed implementare la logica di partizionamento all'interno dei moduli sviluppati per il caricamento dei dati. In quest'ultimo caso va affrontata un'analisi costi / benefici valutando i tempi di sviluppo, l'affidabilità del sistema, la robustezza etc. Considerata la mole dei dati da gestire, l'importanza che gli stessi ricoprono nell'ambito della ricerca, del monitoraggio e della sorveglianza, gli autori del sistema *TSDSystem* hanno sviluppato le logiche di partizionamento all'interno dei moduli di inserimento al fine di mantenere il totale controllo delle tabelle dati, evitando quindi di far gestire all'RDBMS un'operazione reputata cruciale nello sviluppo del sistema.

Da un punto di vista implementativo, il problema è stato affrontato con l'aggiunta dell'entità *measure_link_table* (Figura 2c) i cui attributi principali sono *tableName*, *bTime* ed *eTime* rispettivamente. Questi attributi sono relativi al nome della tabella ed all'intervallo temporale a cui si riferisce. Mediante questo paradigma, la richiesta dell'intervallo temporale viene prima risolta a livello della *measure_link_table*, ottenendo il nome della relativa tabella; la serie temporale risultante sarà ottenuta dall'unione delle *data_Table* cui l'intervallo temporale richiesto si riferisce.

Adottando come criterio selettivo la possibilità di utilizzare un RDBMS open source con licenza *GNU GPL* [GNU documentation] disponibile per diversi sistemi operativi (Windows, Linux, etc...), si è scelto di realizzare il livello database del sistema *TSDSystem* mediante impiego dell'RDBMS open source *MySQL* [MySQL ref.].

3. I Loader

I *loader* sono dei software realizzati al fine di caricare i dati all'interno del database di *TSDSystem*. Questo approccio consente di organizzare i dati secondo il modello proposto in Figura 2 indipendentemente dal formato con cui i dati stessi vengono acquisiti. Nella sua forma più generale, il software *loader* è concepito con l'obiettivo di tradurre un qualsiasi formato in ingresso nella struttura data impiegata per la standardizzazione del dato all'interno di *TSDSystem*. Nella pratica risulta impossibile definire il generico *loader*: questo ostacolo deriva dal fatto che i formati dei dati possono essere di qualsiasi natura. Al fine di realizzare un modulo di caricamento il più generico possibile, sono state avanzate delle ipotesi di progetto, considerando i possibili file contenenti i dati da caricare. I formati considerati al momento della realizzazione del sistema sono file ASCII e origine dati ODBC rispettivamente. Per quanto riguarda i file ASCII, si è considerato il caso in cui i dati prodotti da un *datalogger* o da un generico processo produttore (ad esempio software di elaborazione) siano organizzati in colonne contenenti il tempo in cui il dato è acquisito e le grandezze acquisite. Una tipica struttura dati su file ASCII come quella sopra descritta è riportata in Figura 3. Un altro tipico formato in cui si possono presentare i dati è sotto forma di file *csv* (*Comma Separated Values*). Quest'ultimo è un tipo di file ASCII i cui valori, come esplica l'acronimo stesso, sono separati da una virgola ("," - ma non sono pochi gli esempi in cui si possano incontrare anche altri caratteri di separazione come ";", " ", "TAB", ...). Da quanto detto, queste tipologie di file presentano delle caratteristiche in comune, in particolare:

1. sono organizzati in righe e colonne;
2. i valori delle righe sono separati da un carattere separatore;
3. una colonna è adibita ad archiviare l'informazione temporale.

Alla luce di queste considerazioni, il software *loader* è stato progettato in modo da riconoscere e tradurre nel formato definito per *TSDSystem* qualsiasi struttura del tipo descritto. La corretta lettura dei dati in ingresso viene resa possibile specificando, mediante un opportuno file di configurazione, a quali informazioni fanno riferimento le colonne del file in ingresso. Questa funzionalità è stata implementata

| Date | Value1 | Value2 | ValueN |
|--------------------------|--------|--------------|----------|
| 00.00.050,2577030,592009 | | | 0,327050 |
| 00.00.150,2339870,603753 | | | 0,441090 |
| 00.00.250,3726530,789779 | | | 0,640896 |
| 00.00.350,2504350,835686 | | | 0,411573 |
| 00.00.450,2149600,596562 | | | 0,502425 |
| 00.05.350,2640780,709557 | | | 0,462294 |
| 00.05.450,0998620,666302 | | | 0,493887 |

Al fine di chiarire il concetto, di seguito riportiamo un esempio di riga di un file contenente i dati da interpretare:

Per garantire la corretta interpretazione delle colonne dei dati e l'individuazione delle righe non corrette, si costruisce un'espressione regolare che possa riconoscere una riga come quella dell'esempio. In questo caso una possibile espressione regolare è rappresentata da:

Le espressioni regolari tra parentesi tonde indicano dei gruppi. Ogni gruppo si riferisce ad una particolare colonna di dati, che può essere la data oppure un dato numerico. Il primo gruppo '(d\d\d\d-\d\d-\d\d \d\d:\d\d:\d\d)' effettuerà un match con la parte iniziale della riga '2013-04-23 08:52:00'. Nel file di configurazione si indicherà che la data si trova nella prima colonna.

Il *loader*, mediante quanto descritto, importerà nei database la coppia valore-data (di ogni gruppo sulla tabella di pertinenza) secondo il modello descritto nel paragrafo 2.

Considerata l'elevata quantità di parametri che il generico modulo *loader* deve tenere in considerazione, il file di configurazione presenta una struttura piuttosto complessa. Schematicamente le informazioni si possono dividere in 3 blocchi principali:

1. connessione alle banche dati;
2. definizione del tipo di dato;
3. modalità di inserimento nella struttura *TSDSystem*.

Il primo blocco contiene tutte le impostazioni che riguardano la connessione ai database. In particolare: parametri di connessione al database stazioni ed i parametri di connessione ai database di *TSDSystem*. Il blocco relativo alla definizione del tipo dato conterrà tutte le impostazioni per la corretta interpretazione dell'origine dati (definizione delle tabelle in caso di connessione via ODBC; espressione regolare in caso di lettura da file ASCII). L'ultimo gruppo, relativo alla modalità di inserimento, riporterà i relativi parametri; ad esempio l'intervallo in secondi che ogni volta il *loader* dovrà attendere prima di effettuare una nuova scansione della sua origine dati e avviare il caricamento in *TSDSystem*, quanti dati deve considerare nella fase di lettura (parametro dipendente dalla frequenza di aggiornamento dell'origine dati), il tipo di partizionamento temporale da effettuare in *TSDSystem* (es. partizionamento mensile, settimanale, annuo). L'interfaccia grafica del generico modulo *loader* è riportata in Figura 4. Il colore delle righe è associato ai diversi livelli di criticità riscontrati durante il caricamento dei dati. Questi verranno descritti più in dettaglio nel paragrafo 3.1.



Figura 4. Interfaccia grafica del programma *loader*.

3.1 Stato di funzionamento dei *loader*: il sistema *heartbeat*

Al fine di monitorare lo stato di funzionamento di un processo, spesso si implementano delle cosiddette funzioni di *heartbeat* (letteralmente “battito cardiaco”). Lo scopo di queste procedure è di segnalare se un processo o, più in generale, i *thread* di un processo, sono in esecuzione o meno. L'*heartbeat* può essere visto come un messaggio inviato, con una cadenza prefissata, che segnala ad un sistema di controllo se i processi monitorati risultano attivi. Questo tipo di informazione risulta utile al fine di realizzare un sistema di diagnostica capace di monitorare tutte le parti di un sistema articolato, come nella fattispecie quello oggetto del presente rapporto tecnico. Il sistema di diagnostica implementato in *TSDSystem* è stato realizzato mediante l'impiego di un database in cui sono registrati i *loader* ed i relativi messaggi di *heartbeat*.

Il software *loader* è stato realizzato in modo da archiviare con una cadenza prefissata il messaggio di *heartbeat* nel database. Lo schema E-R del database è riportato in Figura 5. I messaggi di *heartbeat* vengono codificati mediante un codice e una descrizione (attributi *wCode* e *wDescription* rispettivamente); il codice è

relativo alla tipologia di messaggio inviato mentre la descrizione riporta il messaggio in formato testo (leggibile dall'operatore umano) dell'eventuale eccezione (errore di sistema o altro) generato dal gestore delle eccezioni del *loader*. I vari tipi di codice sono stati inseriti in una opportuna tabella e fanno riferimento a delle macro-descrizioni; troviamo ad esempio il codice associato ad un errore di mancata connessione alla rete, uno associato all'errore di lettura del file, uno all'errore di connessione al database etc. Come descritto sopra, il particolare messaggio inviato dal gestore delle eccezioni del processo in esecuzione verrà archiviato nel campo *wDescription*.

Considerato l'elevato numero di *loader* (allo stato attuale il numero complessivo è 114), nonché il numero dei messaggi che possono inviare al database di *heartbeat*, si è proceduto ad un partizionamento della tabella relativa agli *heartbeat message*. In questo caso, la scelta progettuale perseguita prevede la creazione di differenti tabelle *heartbeat message*, una per ogni *loader* attivo, fissando a priori il numero massimo di record che possono essere memorizzati all'interno di ogni tabella. I software di gestione degli *heartbeat* sono principalmente 3:

1. un'interfaccia web di visualizzazione che codifica, tramite tabelle le cui righe seguono un particolare schema cromatico, lo stato complessivo degli *heartbeat* inviati dai *loader* in esecuzione;
2. un software di controllo che segnala tramite e-mail le anomalie segnalate dai *loader* o il mancato invio di *heartbeat* da parte degli stessi;
3. un software per il controllo dello stato del database degli *heartbeat*.

L'interfaccia web di visualizzazione (Figura 6) sintetizza le informazioni del database *heartbeat* mediante una tabella le cui righe sono riferite alle istanze in esecuzione dei *loader* e le sue colonne agli attributi descrittivi. Più in dettaglio, le colonne riportano la descrizione, il nome, la versione, il tempo dell'ultimo *beat* inviato dal *loader*, il nome dell'host e l'IP in cui l'istanza del *loader* è in esecuzione. Il campo calcolato *Status* visualizza mediante un codice di colori lo stato corrente dell'istanza *loader* a cui fa riferimento. In rosso vengono visualizzati i *loader* che non stanno aggiornando il campo di *beat*, la cui causa può essere il blocco del processo, la non raggiungibilità dell'host in cui il processo è in esecuzione o un errore imprevisto che non ha permesso il corretto aggiornamento dell'*heartbeat*.

Il colore giallo evidenzia un generico stato di anomalia che può essere imputata a cause differenti: ad esempio lo stato *warning* può essere raggiunto in caso di un generico errore nelle attività di caricamento dei dati. In questo caso, facendo click sul campo *warning* si accede alle informazioni relative all'anomalia riscontrata.

Tra le anomalie più comuni troviamo, ad esempio, il mancato aggiornamento della sorgente dati cui il *loader* fa riferimento, una corruzione totale o parziale dell'origine dati; se si considera un file ASCII, la condizione *warning* si raggiunge anche nel caso di una riga contenente caratteri inattesi. Nella tabella dei *warning* (Figura 7) i campi cui l'operatore addetto alla manutenzione deve far riferimento sono *WarningCode* e *Description*.

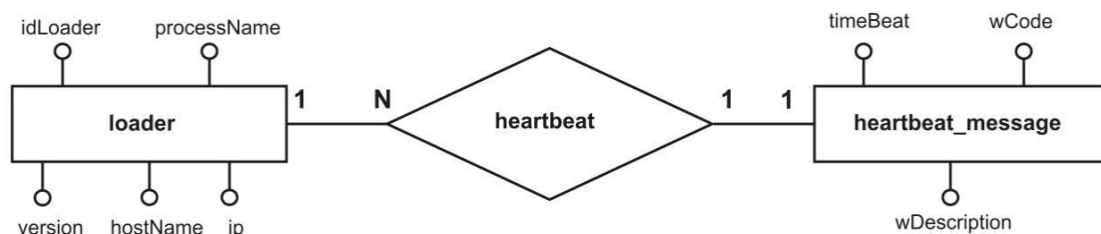


Figura 5. Schema E-R del database *heartbeat*.

General Information

Database Management Staff

Databases

Multiparametric

Seismic

Infrasonic

Clinometric

Voldorad

Radiometer

SO2

GPS Triangle

Saraterm

Magnetic

Meteo (unauthorized)

Data pages

SO2

Radiometer

Monitoring realtime

Realtime S.O.

Realtime Infrasonic Events

Volcanic Tremor Source Location

Special data

SAR Images

Lava flows

KATALOC Earthquakes

OE Earthquakes

CNT Earthquakes

Global Earthquakes

Station Map

Database monitoring

Software monitoring

Software Monitoring

UFS (66)

UFDG (38)

UFVG (6)

UFGM (4)

| Description | Software Name | Version | Last Update UTC | Beat Rate (s) | Host Name | IP | Status |
|--|--------------------|---------|---------------------|---------------|--------------|---------------|---------|
| EPLO_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:54:01 | 60 | adelsis3 | 10.201.1.196 | OK |
| ECNE_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:58 | 60 | adelsis3 | 10.201.1.196 | Warning |
| Last Global Earthquakes Loader | LastGlobalEQ | | 2014-11-07 10:53:58 | 300 | ctdatasismo1 | 10.201.1.206 | OK |
| ECZM_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:55 | 60 | adelsis3 | 10.201.1.196 | OK |
| EPIT_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:55 | 60 | adelsis3 | 10.201.1.196 | OK |
| EBON_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:54 | 60 | adelsis3 | 10.201.1.196 | Warning |
| STR8_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:54 | 60 | adelsis3 | 10.201.1.196 | OK |
| ESVO_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:53 | 60 | adelsis3 | 10.201.1.196 | OK |
| ECPN_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:53 | 60 | adelsis3 | 10.201.1.196 | OK |
| EPON_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:53 | 60 | adelsis3 | 10.201.1.196 | Warning |
| ECPN_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:52 | 60 | adelsis3 | 10.201.1.196 | OK |
| ETFI_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:48 | 60 | adelsis3 | 10.201.1.196 | Warning |
| EBON_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:45 | 60 | adelsis3 | 10.201.1.196 | Warning |
| etna_infra_events_pick seismo_volcanic Acquisition | SignalLoaderDelphi | 3.2 | 2014-11-07 10:53:44 | 60 | adsmelab1 | 10.201.1.215 | Warning |
| ESVO_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:43 | 60 | adelsis3 | 10.201.1.196 | OK |
| Infrasonic Tremor Loader | InfraLoader | | 2014-11-07 10:53:42 | 60 | adelsis3 | 10.201.1.196 | Warning |
| Tremor Loader | TremorLoader | | 2014-11-07 10:53:42 | 180 | adelsis3 | 10.201.1.196 | Warning |
| ESPC_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:41 | 60 | adelsis3 | 10.201.1.196 | Warning |
| Stromboli_trigger stromboli_trigger_hybrid Acquisition | SignalLoaderDelphi | 3.6 | 2014-11-07 10:53:39 | 900 | adelsis3 | 10.201.1.196 | Warning |
| EMNR_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:39 | 60 | adelsis3 | 10.201.1.196 | Warning |
| EMPL_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:36 | 60 | adelsis3 | 10.201.1.196 | OK |
| STR5_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:35 | 60 | adelsis3 | 10.201.1.196 | Warning |
| STR3_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:35 | 60 | adelsis3 | 10.201.1.196 | Warning |
| Last Italian Earthquakes Loader | LastItEq | | 2014-11-07 10:53:31 | 300 | ctdatasismo1 | 10.201.1.206 | OK |
| Meteo Loader | Meteo | | 2014-11-07 10:53:31 | 300 | ctdatasismo1 | 10.201.1.206 | OK |
| ESLN_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:25 | 60 | adelsis3 | 10.201.1.196 | OK |
| EPON_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:24 | 60 | adelsis3 | 10.201.1.196 | Warning |
| EMSG_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:24 | 60 | adelsis3 | 10.201.1.196 | OK |
| ETFI_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:23 | 60 | adelsis3 | 10.201.1.196 | Warning |
| STRB_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:22 | 60 | adelsis3 | 10.201.1.196 | Warning |
| STRE_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:21 | 60 | adelsis3 | 10.201.1.196 | OK |
| STRA_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:20 | 60 | adelsis3 | 10.201.1.196 | Warning |
| EMCN_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:20 | 60 | adelsis3 | 10.201.1.196 | Warning |
| EMFS_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:20 | 60 | adelsis3 | 10.201.1.196 | OK |
| EBEL_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:18 | 60 | adelsis3 | 10.201.1.196 | Warning |
| etna_infra_events_loc seismo_volcanic Acquisition | SignalLoaderDelphi | 3.2 | 2014-11-07 10:53:11 | 60 | adsmelab1 | 10.201.1.215 | Warning |
| IVPL_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:10 | 60 | adelsis3 | 10.201.1.196 | OK |
| EBEM_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:10 | 60 | adelsis3 | 10.201.1.196 | OK |
| STRD_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:10 | 60 | adelsis3 | 10.201.1.196 | OK |
| STR9_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:10 | 60 | adelsis3 | 10.201.1.196 | OK |
| EMFO_infra tremor_nw O Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:10 | 60 | adelsis3 | 10.201.1.196 | Warning |
| STR1_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:09 | 60 | adelsis3 | 10.201.1.196 | Warning |
| ECBO_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:08 | 60 | adelsis3 | 10.201.1.196 | OK |
| EPLO_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:08 | 60 | adelsis3 | 10.201.1.196 | OK |
| ESLN_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:06 | 60 | adelsis3 | 10.201.1.196 | OK |
| ECNE_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:04 | 60 | adelsis3 | 10.201.1.196 | Warning |
| EMFO_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2014-11-07 10:53:04 | 60 | adelsis3 | 10.201.1.196 | Warning |
| Vulkano_trigger vulkano_trigger_hybrid Acquisition | SignalLoaderDelphi | 3.6 | 2014-11-07 10:52:28 | 900 | adelsis3 | 10.201.1.196 | Warning |
| Tremor Classification on ESPC | KKAnalysis-ESPC | | 2014-11-07 10:51:36 | 600 | aqdatames | 10.200.100.79 | OK |
| etna_tremor_loc tremor_loc Acquisition | SignalLoaderDelphi | 3.5 | 2014-11-07 10:41:01 | 900 | adsmelab1 | 10.201.1.215 | Warning |
| ECPN_VLP_Seism sv_events Acquisition | SignalLoaderDelphi | 1.0 | 2014-11-06 17:07:23 | 150000 | adelsis3 | 10.201.1.196 | Warning |
| ECPN_LP_Seism sv_events Acquisition | SignalLoaderDelphi | 1.0 | 2014-11-06 17:07:22 | 150000 | adelsis3 | 10.201.1.196 | Warning |
| EBON_VLP_infra sv_events Acquisition | SignalLoaderDelphi | 1.0 | 2014-10-20 08:48:04 | 150000 | adelsis3 | 10.201.1.196 | Stopped |
| EBEL_LP_infra sv_events Acquisition | SignalLoaderDelphi | 1.0 | 2014-10-20 08:48:03 | 150000 | adelsis3 | 10.201.1.196 | Stopped |
| HMM Etna Fountain Monitoring | HMM Viewer | 1.0 | 2014-07-01 13:17:27 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| SALB wunderground Acquisition | SignalLoaderDelphi | 3.3 | 2014-01-10 11:16:38 | 60 | ctdatasismo1 | 10.201.1.206 | Stopped |
| MASA wunderground Acquisition | SignalLoaderDelphi | 3.3 | 2013-12-27 15:31:34 | 60 | ctdatasismo1 | 10.201.1.206 | Stopped |
| HMM_NOS hmm_fountain_probs Acquisition | SignalLoaderDelphi | 1.0 | 2013-07-16 09:03:02 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| HMM_EQ hmm_fountain_probs Acquisition | SignalLoaderDelphi | 1.0 | 2013-07-16 09:02:59 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| HMM_FOUNTAIN hmm_fountain_probs Acquisition | SignalLoaderDelphi | 1.0 | 2013-07-16 09:02:58 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| HMM_POST hmm_fountain_probs Acquisition | SignalLoaderDelphi | 1.0 | 2013-07-16 09:02:53 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| HMM_QUIET hmm_fountain_probs Acquisition | SignalLoaderDelphi | 1.0 | 2013-07-16 09:02:36 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| HMM_PRE hmm_fountain_probs Acquisition | SignalLoaderDelphi | 1.0 | 2013-07-16 09:02:28 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| EBEL_sismo tremor_nw Z Acquisition | SignalLoaderDelphi | 2.0 | 2013-05-16 07:12:55 | 60 | adelsis3 | 10.201.1.196 | Stopped |
| Software for Testing Purpose | Dummy Software | 1.0 | 2013-04-16 14:30:31 | 600 | adelsis3 | 10.201.1.196 | Stopped |
| Tremor Classification | KKAnalysis-EBEL | | 2013-03-04 07:33:18 | 600 | aqdatames | 10.200.100.79 | Stopped |

Figura 6. Pagina web per la visualizzazione dell’heartbeat; l’ultimo campo della tabella è relativo allo stato.

ESLN_sismo tremor_nw Z Acquisition

1

| Software Name | Beat Time UTC | Warning Code | Description | Status |
|--------------------|---------------------|--------------|--------------------|--------|
| SignalLoaderDelphi | 2014-11-07 11:11:44 | 107 | Zero rows inserted | Error |
| SignalLoaderDelphi | 2014-11-07 11:07:34 | 107 | Zero rows inserted | OK |
| SignalLoaderDelphi | 2014-11-07 10:55:11 | 107 | Zero rows inserted | OK |

Figura 7. Tabella di dettaglio dei messaggi di warning.

Tale approccio permette la completa notifica delle situazioni anomale e il conseguente intervento dell'eventuale personale addetto alla manutenzione del sistema. La tabella principale (Figura 8) riporta una sintesi delle informazioni visualizzabili mediante le pagine web sopra riportate, con l'eccezione della legenda colori.



Il terzo *software* è un applicativo sviluppato in Python (compilato per essere eseguito anche su sistemi operativi Windows), di seguito *hbeatTableChecker*, con l'obiettivo di monitorare lo stato delle tabelle *heartbeat_message*, ed effettuare operazioni di cancellazione dei record quando le dimensioni delle stesse raggiungono una soglia predefinita. L'interfaccia principale del programma è riportata in Figura 9; anche in questo caso le informazioni vengono sintetizzate mediante l'impiego di una tabella riassuntiva dove ogni riga è relativa ad una istanza dei *loader*.

Le colonne riportano rispettivamente il nome della tabella del database istanza dell'entità *heartbeat_message* in cui il *loader* archivia i vari messaggi di *heartbeat*; la colonna *NumRows* riporta il numero di righe (quindi di messaggi) presenti nella tabella; la terza colonna, denominata *MaxRows*, visualizza il numero massimo di righe consentite; la colonna *Percentage* riporta mediante una *progress bar* la percentuale di occupazione della tabella; la colonna *Messages* visualizza l'ultima notifica di controllo sulla tabella come, ad esempio, l'eventuale raggiungimento del numero massimo di record ammissibile; l'ultima

colonna, *Status*, è invece un pulsante che l'utente può cliccare al fine di abilitare o disabilitare il controllo della tabella a cui la riga fa riferimento.

Il programma *hbeatTableChecker*, una volta mandato in esecuzione, esegue un controllo sullo stato di tutte le tabelle segnalando quelle che hanno raggiunto il numero massimo di messaggi consentito e provvede alla re-inizializzazione delle stesse. Dopo la fase di avvio, il programma eseguirà un *check* automatico, con cadenza prefissata di 60 secondi. L'utente può decidere, mediante i pulsanti presenti nella colonna *status*, su quali righe deve essere eseguito il controllo. Uno schema riassuntivo del sistema di *heartbeat* appena descritto è riportato in Figura 10.

The screenshot shows a window titled "Tables Checker connected to (database): heartbeat.tables.exceptions - 2013-05-20 07:42:43". The window contains a table with the following columns: Table name, Numflows, Maxflows, Percentage, Messages, and Status. The table lists 30 different database tables, each with its corresponding flow counts, percentage of messages, and a status button.

| Table name | Numflows | Maxflows | Percentage | Messages | Status |
|---------------------------------------|----------|----------|------------|--|--------|
| CDH_dino_exceptions | 1 | 100 | 1% | No delete from CDH_dino_exceptions 2013-05-23 08:42:40 | [U] |
| CUAD_722_dino_exceptions | 4 | 100 | 4% | No delete from CUAD_722_dino_exceptions 2013-05-23 08:42:40 | [U] |
| CUAD_8262_dino_exceptions | 2 | 100 | 2% | No delete from CUAD_8262_dino_exceptions 2013-05-23 08:42:40 | [U] |
| EBCH_infra_tremor_fw_exceptions | 2 | 100 | 2% | No delete from EBCH_infra_tremor_fw_exceptions 2013-05-23 08:42:40 | [U] |
| EBCH_sismo_tremor_fw_exceptions | 2 | 100 | 2% | No delete from EBCH_sismo_tremor_fw_exceptions 2013-05-23 08:42:40 | [U] |
| EBCH_VLP_infra_eventiv_exceptions | 95 | 500 | 19% | No delete from EBCH_VLP_infra_eventiv_exceptions 2013-05-23 08:42:40 | [U] |
| EBEL_infra_tremor_fw_exceptions | 2 | 100 | 2% | No delete from EBEL_infra_tremor_fw_exceptions 2013-05-23 08:42:40 | [U] |
| EBEL_VLP_infra_eventiv_exceptions | 81 | 500 | 16% | No delete from EBEL_VLP_infra_eventiv_exceptions 2013-05-23 08:42:40 | [U] |
| EBEL_radometer_O_radometer_exceptions | 29 | 100 | 29% | No delete from EBEL_radometer_O_radometer_exceptions 2013-05-23 08:42:40 | [U] |
| EBEL_radometer_exceptions | 11 | 100 | 11% | No delete from EBEL_radometer_exceptions 2013-05-23 08:42:40 | [U] |
| EBEL_radometer_O_radometer_exceptions | 29 | 100 | 29% | No delete from EBEL_radometer_O_radometer_exceptions 2013-05-23 08:42:40 | [U] |
| EBEL_sismo_tremor_fw_exceptions | 57 | 100 | 57% | No delete from EBEL_sismo_tremor_fw_exceptions 2013-05-23 08:42:40 | [U] |
| EBEM_radometer_O_radometer_exceptions | 30 | 500 | 6% | No delete from EBEM_radometer_O_radometer_exceptions 2013-05-23 08:42:40 | [U] |
| EBEM_sismo_tremor_fw_exceptions | 1 | 500 | 0% | No delete from EBEM_sismo_tremor_fw_exceptions 2013-05-23 08:42:40 | [U] |
| ECIO_dino_exceptions | 1 | 100 | 1% | No delete from ECIO_dino_exceptions 2013-05-23 08:42:40 | [U] |
| ECIO_dino_exceptions | 1 | 100 | 1% | No delete from ECIO_dino_exceptions 2013-05-23 08:42:40 | [U] |
| ECIO_sismo_tremor_fw_exceptions | 2 | 100 | 2% | No delete from ECIO_sismo_tremor_fw_exceptions 2013-05-23 08:42:40 | [U] |
| ECIO_VH_dino_exceptions | 2 | 100 | 2% | No delete from ECIO_VH_dino_exceptions 2013-05-23 08:42:40 | [U] |
| ECIO_VH_dino_exceptions | 1 | 100 | 1% | No delete from ECIO_VH_dino_exceptions 2013-05-23 08:42:41 | [U] |
| ECNE_infra_tremor_fw_exceptions | 1 | 100 | 1% | No delete from ECNE_infra_tremor_fw_exceptions 2013-05-23 08:42:41 | [U] |
| ECNE_sismo_tremor_fw_exceptions | 1 | 100 | 1% | No delete from ECNE_sismo_tremor_fw_exceptions 2013-05-23 08:42:41 | [U] |
| ECPR_infra_tremor_fw_exceptions | 1 | 100 | 1% | No delete from ECPR_infra_tremor_fw_exceptions 2013-05-23 08:42:41 | [U] |
| ECPR_VLP_Sesim_eventiv_exceptions | 65 | 500 | 13% | No delete from ECPR_VLP_Sesim_eventiv_exceptions 2013-05-23 08:42:41 | [U] |
| ECPR_sismo_tremor_fw_exceptions | 2 | 100 | 2% | No delete from ECPR_sismo_tremor_fw_exceptions 2013-05-23 08:42:42 | [U] |
| ECPR_VLP_Sesim_eventiv_exceptions | 64 | 500 | 12% | No delete from ECPR_VLP_Sesim_eventiv_exceptions 2013-05-23 08:42:42 | [U] |
| ECPR_sismo_tremor_fw_exceptions | 1 | 500 | 0% | No delete from ECPR_sismo_tremor_fw_exceptions 2013-05-23 08:42:42 | [U] |
| EDAM_dino_exceptions | 2 | 100 | 2% | No delete from EDAM_dino_exceptions 2013-05-23 08:42:42 | [U] |
| EPAL_dino_exceptions | 1 | 100 | 1% | No delete from EPAL_dino_exceptions 2013-05-23 08:42:42 | [U] |
| EPCH_sismo_tremor_fw_exceptions | 1 | 500 | 0% | No delete from EPCH_sismo_tremor_fw_exceptions 2013-05-23 08:42:42 | [U] |
| EPHEG_dino_exceptions | 1 | 100 | 1% | No delete from EPHEG_dino_exceptions 2013-05-23 08:42:42 | [U] |

Figura 9. Finestra principale del software *hbeatTableChecker*.

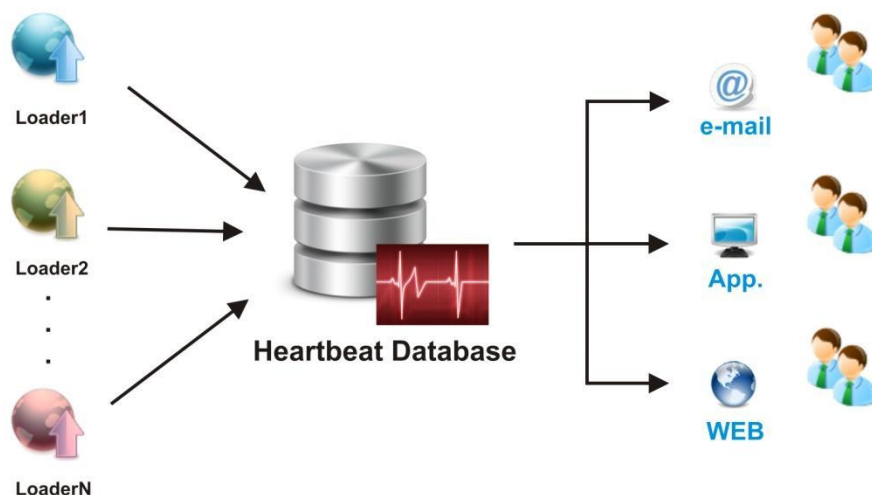


Figura 10. Schema di principio del sistema di *heartbeat*.

3.2 Interfaccia di configurazione del *loader*

Al fine di facilitare la creazione dei file di configurazione per l'inizializzazione dei *loader*, è stato realizzato un opportuno *software* che permette l'inserimento dei parametri mediante una interfaccia grafica. La prima finestra (Figura 11) del programma di configurazione permette l'inserimento dei parametri di connessione relativi al database principale contenente la definizione delle tabelle dati e delle procedure/funzioni per il trattamento degli stessi. Queste serviranno per la creazione del nuovo database a cui il *loader* farà riferimento. La compilazione di questi parametri è necessaria quando un *loader*, adibito al caricamento di una nuova sorgente dati, viene stanziato. I parametri richiesti in questa fase di configurazione sono il nome degli *hosts*, *user* e *password* per l'accesso ai server ed il nome del database destinato ad archiviare i dati della nuova serie temporale. La seconda finestra di configurazione (Figura 12) permette l'impostazione dei diversi parametri che verranno utilizzati dalle istanze dei vari *loader* in esecuzione.

The screenshot shows a Windows-style dialog box titled "Dump DB". It contains two sections: "Source DB Signal Connection Properties" and "Destination DB Signal Connection Properties". Each section has input fields for Host, Port, User, Password, and Database. The Source section has pre-filled values: Host: ctdatasismo1, Port: 3306, User: sismodev, Password: *****, Database: db_signal_skeleton. The Destination section has empty fields for Host, Port, User, Password, and Database. A "Create DataBase" button is located at the bottom right of the Destination section.

Figura 11. Finestra per l'inserimento dei parametri relativi alle connessioni con i database.

The screenshot shows a larger application window titled "FormMain". It has a "Global section" tab selected. The "Ini file path" is set to "C:\Users\robosismov\ Desktop\Signal.loader v.2.0\sismo_tremor_config.ini". Below this is a "Set Default Values" button. The main area is divided into three sections: "Heartbeat Connection", "Recovery Settings", and "PermanentNet Connection". The "Heartbeat Connection" section has fields for Host, User, Password, Database, Software Name, and Version. The "Recovery Settings" section has "Start Recovery Date" and "End Recovery Date" fields with date pickers. The "PermanentNet Connection" section has fields for Host, User, Password, and Database. Red annotations are present: a vertical line 'I' on the left, a vertical line 'II' on the left, and three vertical lines 'III' on the right. The "OK" and "Cancel" buttons are at the bottom right.

Figura 12. Sezione *Global section* del programma di configurazione.

L'interfaccia grafica per l'inserimento dei parametri che serviranno a creare il file di configurazione del *loader*, è organizzata in “sezioni” e “sottosezioni”. I due pulsanti della *toolbar* (pannello I di Figura 12), permettono di caricare un file di configurazione già esistente, e quindi di salvarne poi le modifiche, oppure di creare un nuovo file di configurazione in base alle impostazioni inserite. Le due principali sezioni (Figura 12) sono denominate rispettivamente *Global section* e *Station section* (pannello II); la prima serve per l'inserimento delle connessioni al database degli *heartbeat* (descritto nel paragrafo 3.1) ed al database delle stazioni [Aliotta et al., 2010]. Inoltre è possibile specificare un intervallo temporale, eventualmente da recuperare, impostando la data di inizio e di fine nel pannello *Recovery Settings* (pannello III).

Il dettaglio della sezione denominata *Station section* è riportata in Figura 13. La prima sottosezione, *Data connection*, permette l'inserimento dei parametri per la connessione al database preposto all'inserimento dei dati caricati tramite l'istanza del *loader* che entrerà in esecuzione. Nelle voci riportate nel pannello I, l'utente specifica *Host*, *User*, *Password* e nome del database in cui verranno ubicati i dati acquisiti. Il pannello II permette l'inserimento delle informazioni relative al canale di acquisizione cui la sorgente dati da caricare fa riferimento. Tali informazioni vengono reperite dal database delle stazioni [Aliotta et al., 2010]; le principali voci cui la sorgente dati fa riferimento vengono di seguito elencate:

- *Site_name*: nome del sito;
- *Node_name*: nome del sensore;
- *Sensor_Type*: tipo di sensore;
- *Channel_name*: nome del canale;
- *idChannel*: identificativo univoco del canale all'interno del database delle stazioni;
- *idSignalType*: identificativo relativo al tipo di serie temporale;
- *Column Idx*: colonne delle sorgente dati (file ASCII formattato o tabella origine raggiungibile mediante connessione ODBC) da caricare nel database.

In particolare l'identificativo *idChannel* permette la connessione logica tra il canale e le serie temporali da acquisire all'interno di *TSDSystem*.

Figura 13. Dettaglio della sotto-sezione *Data connection*.

La seconda sottosezione, denominata *heartbeat conf* (Figura 14), permette l'inserimento dei parametri per la connessione al database *heartbeat* descritto nel paragrafo 3.1. Tra i parametri più importanti trattati in questa sezione, vi sono il nome della tabella che conterrà l'elenco di tutti i messaggi di *warning*, il nome della tabella relativa agli attributi dell'istanza del programma *loader* che si sta configurando, il nome dell'*host* in cui il *loader* è in esecuzione, e il valore del *beat rate* inteso come tempo che intercorre tra un *beat* ed il successivo. In particolare, quest'ultimo parametro è fondamentale per il monitoraggio delle attività

del *loader*; se il tempo di attesa tra un *beat* ed il successivo è troppo alto, questo può indicare problemi del programma in esecuzione.

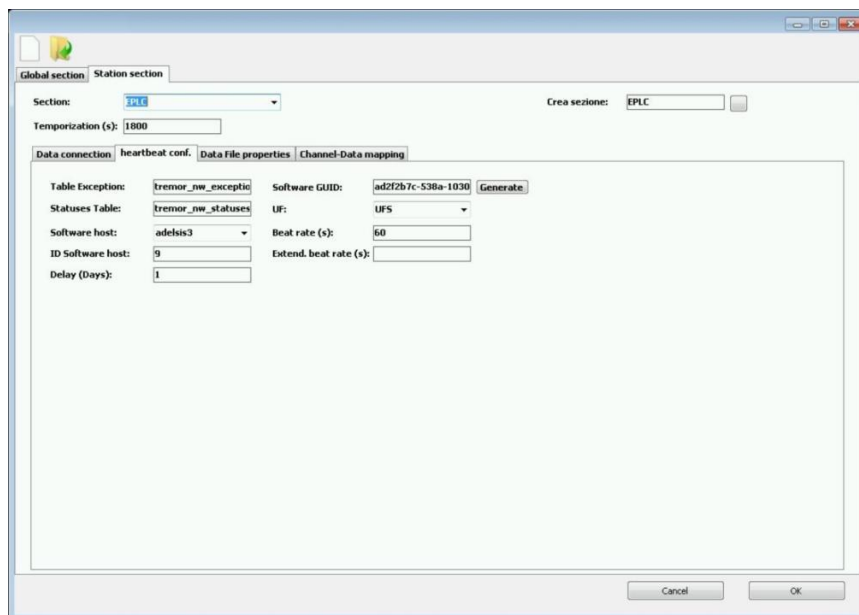


Figura 14. Dettaglio della sotto sezione *heartbeat conf.*

Parte dei parametri sopra descritti, come ad esempio il nome delle tabelle del database, vengono impostati automaticamente dal software di configurazione in quanto sono valori già predefiniti all'interno di *TSDSystem*. Mentre le due sottosezioni precedentemente descritte sono relative principalmente al settaggio dei parametri per la connessione ai database che fanno parte di *TSDSystem*, la sottosezione *Data File Properties* (Figura 15) permette l'inserimento dei parametri per la lettura della sorgente dati. I parametri relativi a questa sezione (che verranno di seguito descritti) sono validi nel caso in cui il programma *loader* ha come sorgente dati un file ASCII formattato. Al momento il programma di configurazione non permette l'inserimento dei parametri per la connessione a sorgenti dati ODBC; questi ultimi pertanto devono essere impostati manualmente dall'operatore in un'apposita sezione del file di configurazione. Tra le voci principali della sezione *Data File Properties* troviamo *RegExp String*, che permette l'impostazione dell'espressione regolare descritta nel paragrafo 3; la voce *url*, per impostare il percorso del file dati; la voce *Lines to read* riferita al numero di righe che ad ogni ciclo di lettura il programma *loader* deve considerare. La voce di tipo *check* denominata *Multiple file* indica al *loader* che l'origine dati è suddivisa in più file: in questo caso il programma di caricamento leggerà sempre dall'ultimo file creato. Tutte le altre voci, di cui per brevità omettiamo la descrizione, sono relative alla descrizione del formato data e tempo (*date* e *time*) da considerare, al nome del file e numero di riga da considerare per l'avvio dei cicli di lettura, al tipo di separatore decimale per l'inserimento dei valori numerici in *TSDSystem* etc. L'ultima voce degna di nota è il *check Directory time system*; se abilitato, questo predispone il programma *loader* a leggere i dati da file organizzati all'interno del *file system* mediante *directory* annidate relative ad anno, mese e giorno rispettivamente.

L'ultima sottosezione del programma di configurazione è *Channel-data mapping* (Figura 16). Questa permette l'inserimento delle voci *Sample Frequency* e *Table Timing* rispettivamente. La prima voce imposta la frequenza di campionamento, se definita, della serie temporale da caricare; la seconda permette di impostare il *time splitting* delle tabelle contenenti i dati. Come spiegato nel paragrafo 2, le tabelle adibite all'archiviazione dei dati possono essere partizionate in base a criteri temporali; nel sistema oggetto del presente report le tabelle possono essere suddivise in tabelle giornaliere, mensili o annuali. La scelta del tempo di partizionamento dipende dalla mole di dati acquisiti. Ad esempio, i dati acquisiti con frequenza giornaliera possono essere archiviati in tabelle annuali mentre, i dati acquisiti con periodo di campionamento dell'ordine dei secondi, possono essere archiviati in tabelle giornaliere o mensili. Come spiegato nel paragrafo 2, questo approccio all'archiviazione dei dati ottimizza la gestione dell'intero sistema.

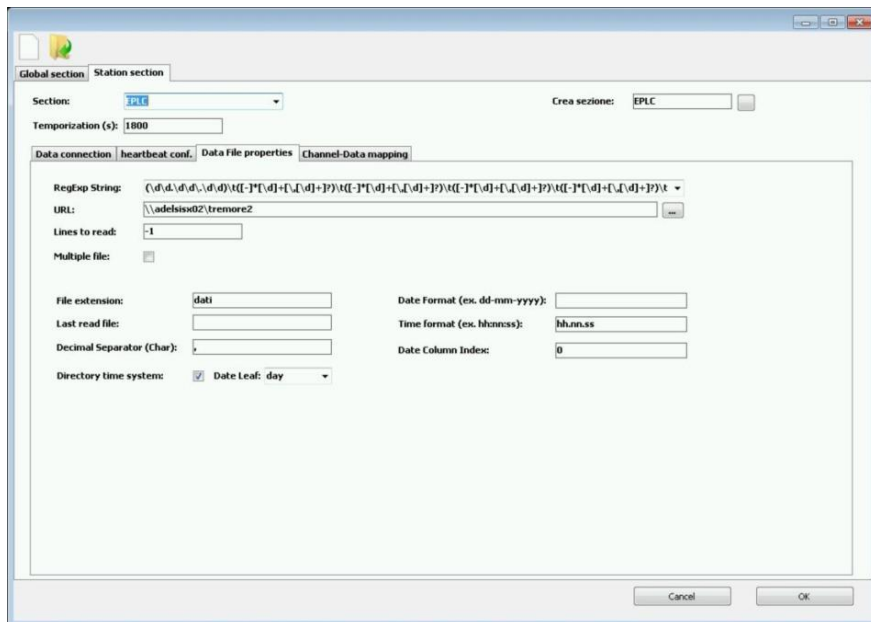


Figura 15. Dettaglio della sottosezione *Data file properties*.

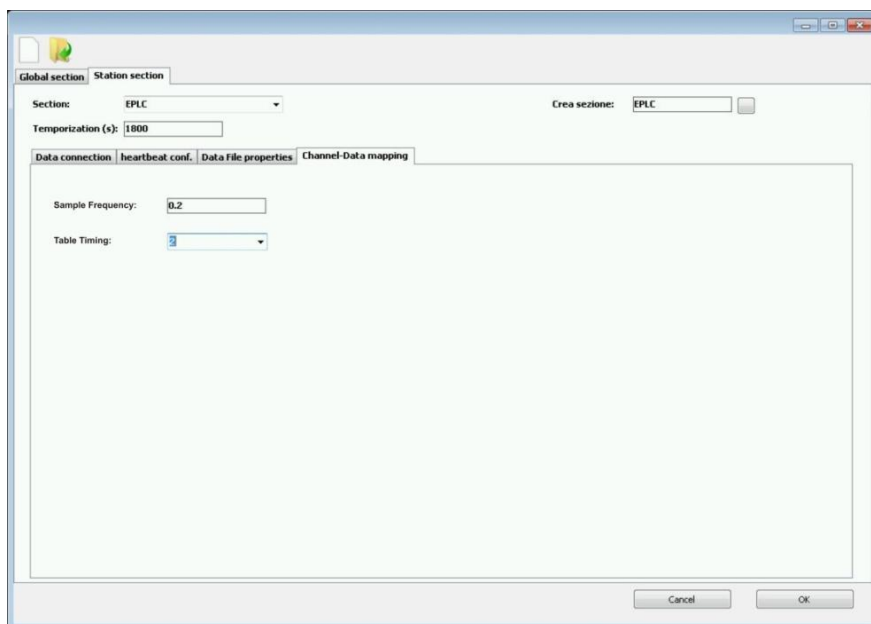


Figura 16. Dettaglio della sottosezione *Channel-data mapping*.

4. Applicativi per la visualizzazione e l'interrogazione dei dati di *TSDSystem*

Oltre ai software realizzati per il caricamento dei dati e per il controllo dello stato dei database che compongono *TSDSystem*, sono stati implementati due differenti sistemi per la visualizzazione delle serie temporali acquisite dal sistema. I moduli di visualizzazione realizzati sono principalmente di 2 tipi:

1. un portale web per la navigazione dei dati di *TSDSystem*;
2. un applicativo *win32* (applicativo per sistemi Microsoft *Windows*) per l'interrogazione e la visualizzazione delle serie temporali archiviate.

Il primo sistema permette agli utenti di scegliere quali serie temporali visualizzare ed il relativo periodo di interesse, utilizzando un portale web; il secondo programma, chiamato *TSViewer*, è invece un software interattivo che gli utenti possono installare sul proprio computer e utilizzare come organo di visualizzazione. Entrambi gli strumenti hanno lo scopo di garantire l'accesso a *TSDSystem* agli utenti che ne fanno richiesta; ciò che cambia è la modalità di accesso. Mentre il portale web offre una serie di funzionalità per la visualizzazione congiunta di serie temporali, utili per scopi di ricerca, l'applicativo *TSViewer* è stato progettato per una modalità di funzionamento H24, inserendosi pertanto in un contesto orientato più a scopi di monitoraggio e sorveglianza. Nei paragrafi seguenti entrambi i software verranno descritti con maggiore dettaglio.

Al fine di gestire l'utenza che effettua richieste al database, il sistema *TSDSystem* implementa la possibilità di gestire i livelli di accesso. Questa funzionalità è stata implementata mediante un database ausiliario. Ad ogni utente presente nel database possono essere assegnati o revocati i diritti di accesso al database delle serie temporali. Mediante questa politica i livelli di accesso possono essere specializzati per ogni singolo utente in base ai dati che questi può visualizzare, scaricare, nonché gli intervalli temporali che può richiedere. Tale approccio permette sia di gestire diversi livelli di utenza che di garantire la sicurezza dei dati inseriti. L'operazione di inserimento e di assegnazione dei permessi viene demandata agli amministratori del sistema *TSDSystem*.

4.1 L'applicativo *TSViewer*

L'applicativo *TSViewer* è stato progettato per la visualizzazione dei dati contenuti all'interno del sistema *TSDSystem*. Una volta mandato in esecuzione, *TSViewer* controlla mediante connessione a *TSDSystem* se l'utente attualmente *loggato* nel computer in cui è in esecuzione l'istanza di *TSViewer* possiede una o più autorizzazioni di accesso al sistema.

TSViewer è stato progettato per funzionare sia in modalità utente che in modalità "monitoraggio": quest'ultima pone il *TSViewer* in aggiornamento automatico su una serie temporale prefissata. L'interfaccia grafica del programma è divisa in 3 parti principali (Figura 17): una toolbar (pannello I), un pannello di visualizzazione grafica (pannello II), e un pannello per la selezione dei sensori, dei canali e degli intervalli temporali (pannello III). Il primo pannello mette a disposizione dell'utente una serie di funzionalità per l'interazione con il grafico visualizzato nel pannello II. Le funzioni dei singoli bottoni sono riportate in tabella 1. Il pannello III permette l'impostazione dei parametri per effettuare le richieste al database. Tra le funzionalità presenti nel pannello I troviamo stato della connessione con *TSDSystem*, funzioni di zoom del grafico, attivazione/disattivazione del cursore mobile, attivazione/disattivazione della modalità di visualizzazione con scala logaritmica delle ordinate, apertura della pagina web di sintesi dello stato di funzionamento degli *heartbeat* (utile per verificare se il *loader* di una determinata serie temporale si è bloccato oppure presenta problemi di caricamento dei dati), attivazione della modalità di monitoraggio, e infine salvataggio del grafico su file immagine jpg.

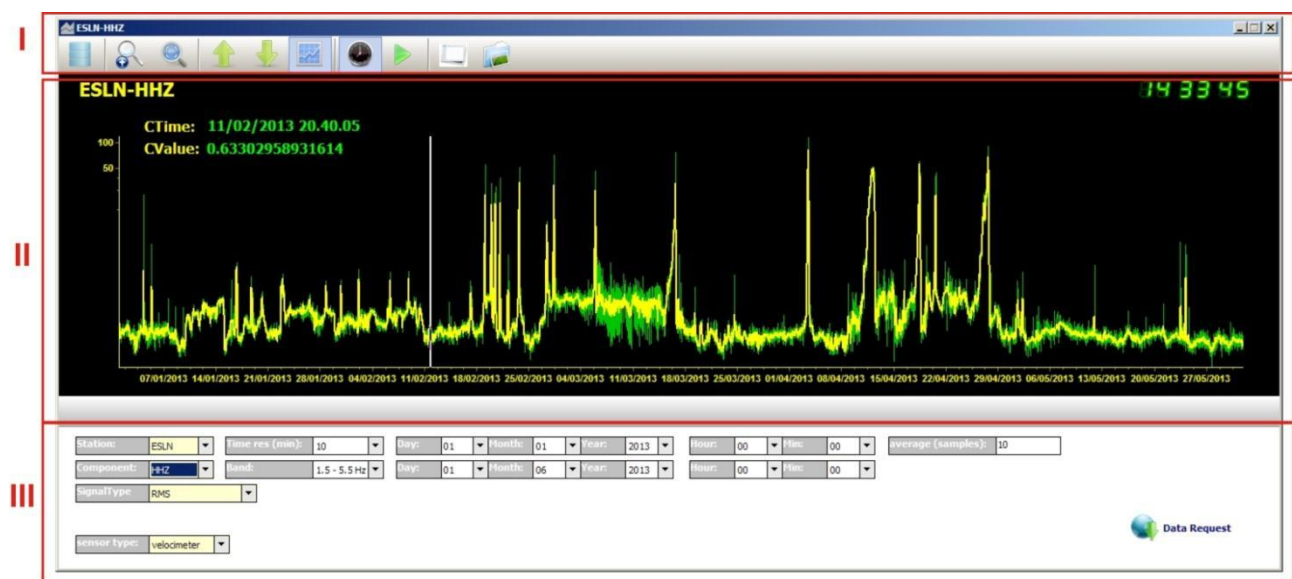


Figura 17. Interfaccia del software *TSViewer*.











| | |
|---|---|
|  | Verifica dello stato di connessione a <i>TSDSystem</i> |
|  | Attivazione dello zoom sul grafico |
|  | Reset dello zoom |
|  | Aumento del limite superiore dell'asse delle ordinate |
|  | Diminuzione del limite superiore dell'asse delle ordinate |
|  | Visualizzazione in scala logaritmica (solo ordinate) |
|  | Attivazione / disattivazione del cursore mobile |
|  | Attivazione / disattivazione della modalità di “monitoraggio” |
|  | Connessione al centro di controllo <i>heartbeat</i> |
|  | Salvataggio del grafico su file immagine <i>.jpg</i> |

Tabella 1. *Toolbar* principale di *TSViewer*.

I parametri ubicati nel pannello III (Figura 18) servono per la richiesta delle serie temporali. Questi sono:

- tipo di sensore;
- nome del sensore;
- canale del sensore;
- tipo di segnale associato al canale.

Come descritto nel capitolo 2, ad ogni canale possono essere associate più serie temporali. Gli altri parametri che l'utente può impostare sono il sottocampionamento della serie richiesta e l'intervallo temporale. Una volta impostati, la richiesta dell'intervallo temporale viene inoltrata mediante click sul pulsante *Data Request*, ubicato sempre nel pannello III. In alternativa, l'utente può attivare la modalità “monitoraggio” mediante l'apposito pulsante della *toolbar* di cui al pannello I (Figura 17).

Per attivare quest'ultima modalità non è necessaria l'impostazione di nessun intervallo temporale: il visualizzatore richiederà in automatico al database gli ultimi 7 giorni della serie temporale desiderata e provvederà ad un aggiornamento automatico ogni 60 secondi. Questi valori possono essere eventualmente impostati nel file di configurazione di *TSViewer*.

A titolo di esempio, consideriamo il caso in cui un utente desidera accedere a *TSDSystem* per visualizzare serie temporali concernenti i dati sismici. Se l'utente è abilitato, una volta lanciato il programma sarà possibile impostare i parametri di ricerca dal pannello III. In Figura 18a è riportato un dettaglio delle caselle di testo combinate (*combobox*) dove l'utente può impostare i parametri di ricerca relativi a tipo di sensore, stazione, componente e tipo di dato archiviato. Nell'esempio riportato è stato selezionato un sensore di tipo *velocimeter*, la stazione (*station*) denominata “ECPN”, il canale di acquisizione (*component*) “HHZ” e la serie temporale (*SignalType*) “RMS”. Come riportato in Figura 18b, è possibile selezionare diversi tipi di serie temporali associate al canale HHZ di ECPN: *n_events_lp* (numero di eventi sismo-vulcanici di tipo LP), *mean_ampl_lp* (ampiezza media degli eventi), etc. Oltre all'impostazione dei suddetti parametri, nel pannello III sono presenti i controlli per l'impostazione degli intervalli temporali, del sotto-campionamento e della media mobile (Figura 18c). Mediante il pulsante “Data Request” (in basso a destra nel pannello III), la richiesta dati viene inviata a *TSDSystem*. Dopo un intervallo di caricamento (*buffering*) del dato, il risultato sarà visualizzato nell'apposito pannello (pannello II della Figura 17). Mediante le funzioni della *toolbar* (Tabella I), l'utente può attivare diverse funzionalità utili all'esplorazione del dato quali zoom, cursore mobile, esportazione del grafico e scelta del tipo di scala temporale. Le fasi della richiesta del dato sono riportate in Figura 19.

Facendo click con il pulsante destro del mouse è inoltre possibile visualizzare un menu contestuale contenente funzionalità aggiuntive, quali la possibilità di visualizzare congiuntamente una serie temporale insieme alla relativa media mobile, solo serie temporale o solo media mobile. Queste opzioni permettono all'utente di personalizzare il tipo di grafico visualizzato.

a) Station: ECPN
Component: HHZ
Signalttype: RMS
sensor type: velocimeter

b) Station: ECPN
Component: HHZ
Signalttype: RMS
sensor type: velocimeter

c) Time res (min): 10
Band: 1.5 - 5.5 Hz
Day: 01 Month: 01 Year: 2013
Hour: 00 Min: 00
average (samples): 10

Figura 18. Dettaglio del pannello III di *TSViewer*.

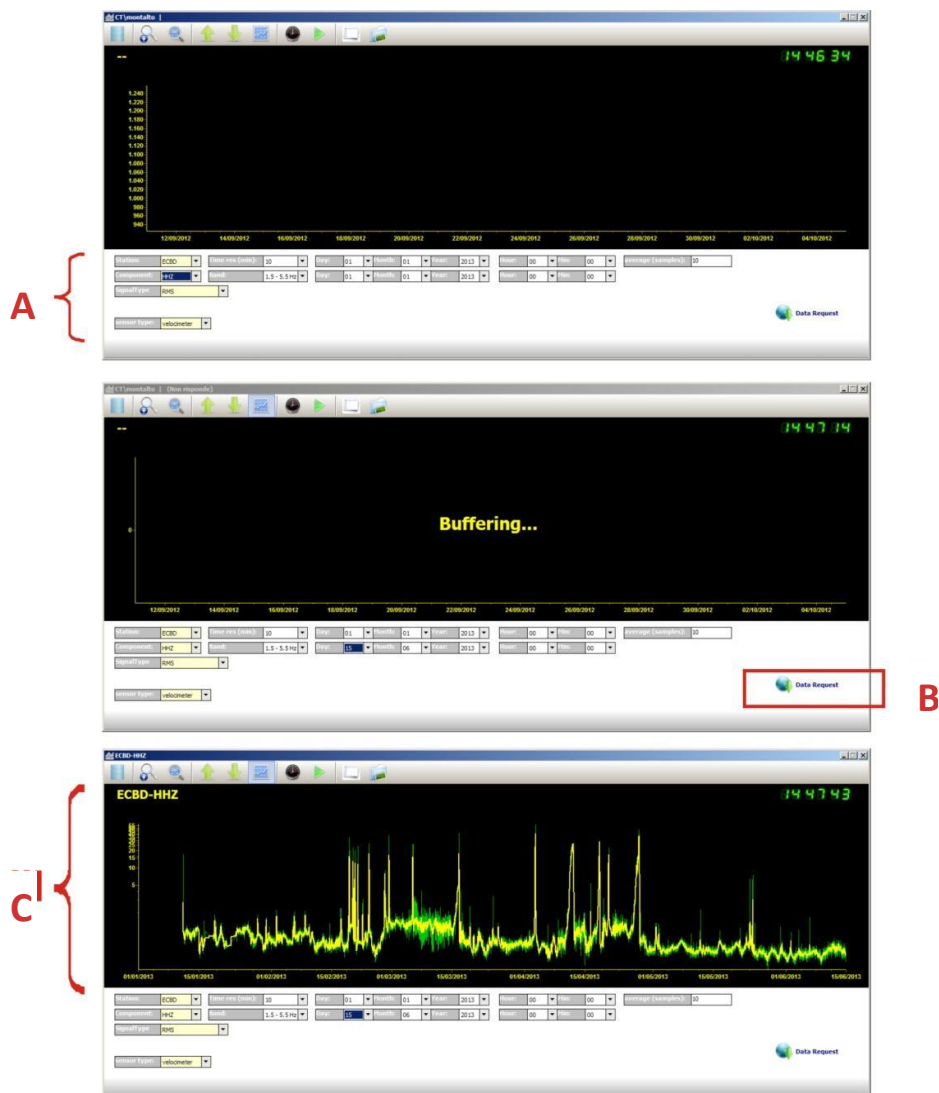


Figura 19. Richiesta del dato: A) Impostazione dei parametri nel pannello III; B) invio della richiesta a *TSDSystem*; C) la serie temporale è visibile nel pannello II.

4.2 Utilizzo di *TSViewer* per il monitoraggio

L'applicativo *TSViewer*, oltre ad essere utilizzato come interfaccia di interrogazione al sistema *TSDSystem*, può essere impiegato come visualizzatore in *realtime* delle serie temporali acquisite. Al momento della stesura del presente report, l'applicativo *TSViewer* è impiegato sia come software per la visualizzazione delle serie temporali del tremore vulcanico nella Sala Operativa dell'Osservatorio Etneo, sia come applicativo distribuito agli utenti. Come descritto nel paragrafo 4.1, una volta inseriti i parametri relativi al sensore, al canale e alla serie temporale, la modalità "monitoraggio" può essere abilitata mediante l'apposito pulsante della *toolbar*. In questo caso *TSViewer* eseguirà automaticamente l'aggiornamento della serie temporale visualizzata. Nell'esempio di Figura 20 è riportata la visualizzazione delle serie temporali relative al segnale radar, al segnale sismico ed infrasonico.

Differenti colori individuano differenti serie temporali; dall'alto in basso sono visualizzate rispettivamente le serie temporali relative a:

1. segnale acquisito dal radar ubicato al sito "La Montagnola" (EMON);
2. RMS segnale sismico acquisito al sito "Bocca Nuova" (componente verticale EBCN);
3. RMS segnale infrasonico acquisito al sito "Bocca Nuova" (componente infrasonica EBCN);
4. RMS segnale sismico acquisito al sito "Serra La Nave" (componente verticale ESLN);
5. RMS segnale infrasonico acquisito al sito "Serra La Nave" (componente infrasonica ESLN);
6. RMS segnale sismico acquisito al sito "Serra Pizzuta Calvarina" (componente verticale ESPC);
7. RMS segnale sismico acquisito al sito "Torre del Filosofo" (componente verticale ETFI);
8. RMS segnale infrasonico acquisito al sito "Torre del Filosofo" (componente infrasonica ETFI);
9. RMS segnale sismico acquisito al sito denominato STR1 ubicato a Stromboli;
10. RMS segnale sismico acquisito al sito denominato STR8 ubicato a Stromboli;
11. RMS segnale sismico acquisito al sito IVPL ubicato a Vulcano Piano (isola di Vulcano).



Figura 20. Applicativo *TSViewer* in funzione nella Sala Operativa dell'Osservatorio Etneo. I differenti colori delle serie sono relative alla tipologia di dato visualizzata; in azzurro il segnale del radar ubicato al sito La Montagnola; in verde l'ampiezza del tremore vulcanico; in rosso l'RMS del segnale infrasonico; in blu l'RMS del segnale sismico relativo a tre stazioni delle Eolie.

Al fine di gestire differenti istanze di *TSViewer* in un contesto di visualizzazione avanzata, come quella richiesta in una sala di monitoraggio, si è realizzato un modulo software per la gestione automatica di uno o più visualizzatori che funzionano simultaneamente (Figura 21). Questa modalità di gestione permette la visualizzazione congiunta di più serie temporali mediante l'utilizzo di uno o più monitor. L'applicativo per la gestione dei visualizzatori, di seguito *TSViewerMonitor*, è riportato in Figura 21. L'interfaccia grafica principale è logicamente suddivisa in 3 pannelli: I) *toolbar* contenente i bottoni adibiti al caricamento dei file di configurazione e all'avvio del sistema di visualizzazione, II) Elenco delle stazioni relative alle serie temporali da visualizzare, III) *console* per la visualizzazione dei comandi eseguiti da *TSViewerMonitor*. In quest'ultimo pannello vengono riportati, con frequenza di aggiornamento prefissata, l'elenco delle operazioni eseguite ed il loro esito; *TSViewerMonitor* controlla lo stato delle istanze dei visualizzatori riportando nella *console* il loro stato (processo in esecuzione o bloccato), le eventuali operazioni di riavvio eseguite, nonché eventuali errori riscontrati nella comunicazione con i visualizzatori stessi. L'operazione principale che *TSViewerMonitor* esegue è il controllo del corretto funzionamento di ogni singola istanza di *TSViewer*: se la singola istanza non risponde (processo bloccato) questa verrà chiusa e riavviata dopo un intervallo di tempo predefinito. Questa funzionalità preserva il sistema da eventuali malfunzionamenti dovuti sia ai software di visualizzazione stessi che a problemi legati al sistema operativo. Mediante questo accorgimento, ogni volta che i processi di visualizzazione si bloccano vengono riavviati senza l'intervento dell'operatore umano. Le operazioni, oltre ad essere visualizzate nella *shell* del pannello III, vengono registrate in un file di log. Quest'ultimo si rivela utile agli amministratori di *TSDSystem* per generare delle statistiche di funzionamento delle varie parti che compongono il sistema (nel caso in esame i log sono riferiti ai visualizzatori), in modo da avere un quadro completo sulle diverse anomalie, o malfunzionamenti in genere, che possono affliggere il sistema.

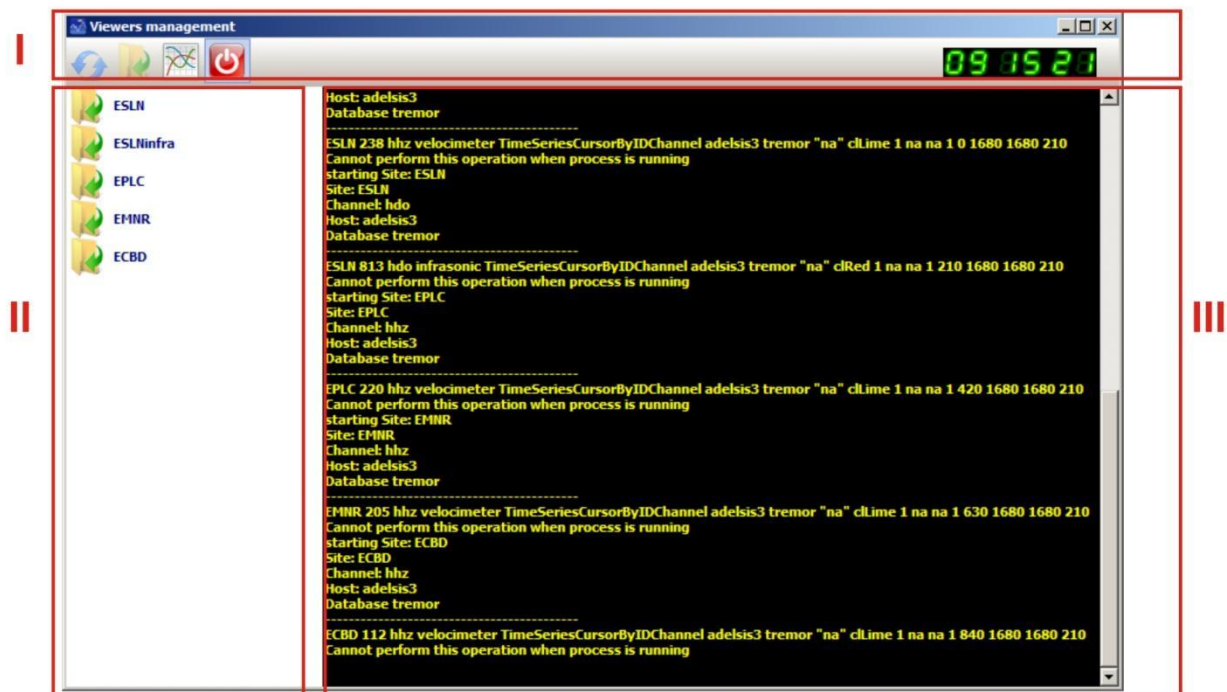


Figura 21. Interfaccia grafica dell'applicativo *TSViewerMonitor*.

Un tipico file di configurazione di *TSViewerMonitor* è riportato in Figura 22. La struttura è organizzata per fornire al programma *TSViewerMonitor* le informazioni necessarie per l'avvio e la gestione dei visualizzatori. Tra parentesi quadre troviamo il nome della sezione: ogni sezione è relativa ad un singolo visualizzatore, pertanto più sezioni contengono i parametri di avvio di più visualizzatori; i nomi delle sezioni possono essere dati arbitrariamente. I parametri della singola sezione sono riportati in tabella II.

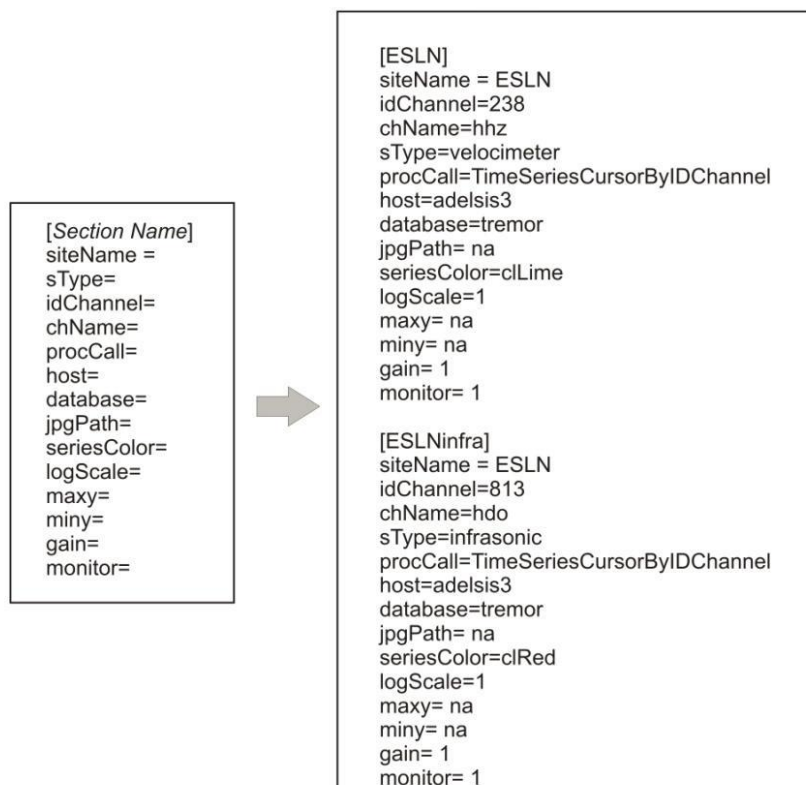


Figura 22. file di configurazione dell'applicativo *TSViewerMonitor*.

| | |
|--------------------|---|
| <i>siteName</i> | nome del sito a cui il visualizzatore fa riferimento |
| <i>sType</i> | tipo di sensore |
| <i>idChannel</i> | identificativo numerico del canale di acquisizione a cui la serie temporale è logicamente connessa |
| <i>chName</i> | nome del canale |
| <i>procCall</i> | nome della procedura di <i>TSDSystem</i> da invocare per l'accesso alla serie temporale |
| <i>host</i> | nome dell' <i>host</i> (macchina <i>server</i>) in cui è installato il database |
| <i>database</i> | nome della porzione di database dove è archiviata la serie temporale richiesta |
| <i>jpgPath</i> | nome del percorso dove andranno memorizzati, se la funzionalità è abilitata, i grafici in formato <i>jpg</i> della serie temporale visualizzata |
| <i>seriesColor</i> | colore della serie temporale da visualizzare |
| <i>logScale</i> | parametro booleano per attivare disattivare la scala logaritmica sull'asse delle ordinate del grafico |
| <i>maxy</i> | se impostato fissa l'estremo superiore dell'asse delle ordinate |
| <i>miny</i> | se impostato fissa l'estremo inferiore dell'asse delle ordinate |
| <i>gain</i> | valore di amplificazione da assegnare alla serie temporale da visualizzare |
| <i>monitor</i> | Valore numerico (0..n) che indica il monitor su cui visualizzare la serie temporale |

Tabella 2. descrizione dei parametri di *TSViewerMonitor*.



Figura 23. Pagina web per la visualizzazione dei grafici del tremore vulcanico. La pagina si aggiorna automaticamente ogni 60 secondi.

4.3 Interfaccia web per la visualizzazione delle serie temporali

Come accennato nel paragrafo 4.2, i singoli visualizzatori in modalità “monitoraggio” possono generare automaticamente degli *snapshot* automatici su file immagine *jpg* delle serie temporali visualizzate. Questa opzione risulta essere utile per la condivisione dei grafici mediante l’impiego di pagine web aggiornate automaticamente. Questa particolare funzionalità è impiegata, ad esempio, per visualizzare i grafici relativi al tremore all’interno della pagina web accessibile dal sito internet di sezione <http://www.ct.ingv.it/it/tremore-vulcanico.html> (Figura 23). Questa opzione permette la visualizzazione non interattiva dei grafici relativi ad una finestra temporale fissa. Nell’esempio di Figura 23 l’intervallo temporale considerato è di 7 giorni.

Al fine di rendere i dati fruibili mediante un portale web intranet/internet, è stata realizzata un’applicazione web che permette sia l’interrogazione dei dati presenti in *TSDSystem*, che l’interazione dell’utente con i grafici mediante funzioni di *zoom*, sincronizzazione tra le diverse serie temporali visualizzate, cursore mobile etc. La pagina principale del portale è riportata in Figura 24. Questa presenta un pannello principale che ospita una visualizzazione multipla di serie temporali mediante l’impiego di grafici interattivi, ed un pannello laterale (pannello I di Figura 24) per l’apertura delle pagine di ricerca. I grafici

interattivi (pannello II di Figura 24) forniscono all'utente funzioni di *zoom* e di *picking* temporale mediante un cursore mobile. Nel caso di visualizzazione multipla, lo *zoom* eseguito su una serie temporale sincronizza tutte le altre serie visualizzate nella pagina, in modo che esse presentino tutte lo stesso fattore di *zoom*.

Questa funzione permette la navigazione simultanea e sincronizzata dei dati visualizzati.

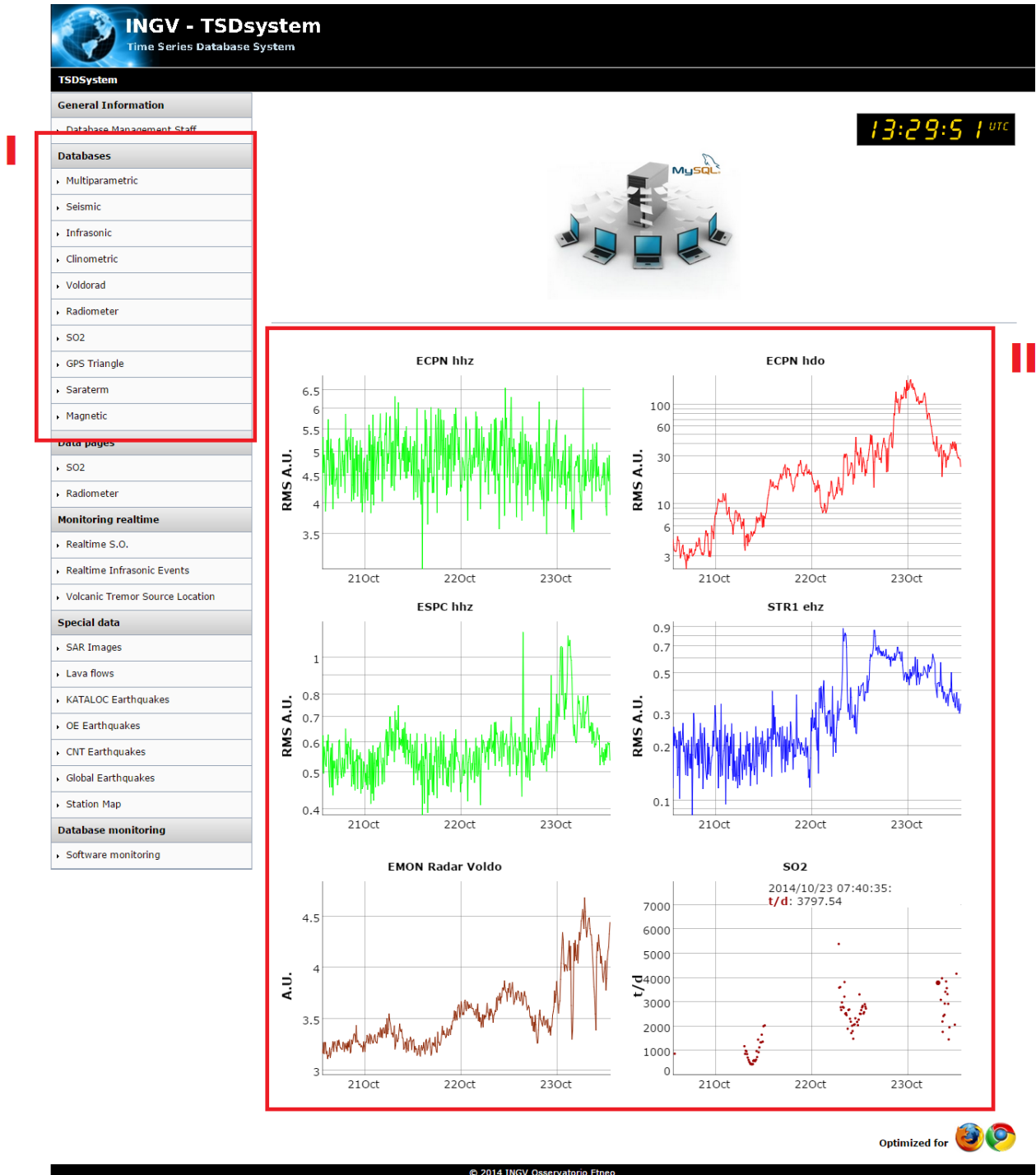


Figura 24. Pagina web per l'interrogazione e la visualizzazione delle serie temporali archiviate in *TSDsystem*.

La scelta del tipo di serie temporale, il numero complessivo delle serie e l'intervallo temporale da visualizzare vanno impostati mediante un opportuno file di configurazione a cui le pagine web presenti sul server fanno riferimento. Anche in questo caso, ogni singolo grafico si aggiorna automaticamente ad intervalli di tempo prefissati.

L'aspetto più importante dell'applicazione web è la possibilità di effettuare ricerche all'interno di *TSDSystem*. L'accesso ai motori di ricerca relativi alle singole tipologie di dato (dati sismici, clinometrici, infrasonici etc.) avviene cliccando sui link riportati nel pannello I della pagina web principale (Figura 24). Sempre dallo stesso menu è possibile attivare un motore di ricerca multiplo per la richiesta e visualizzazione congiunta di più serie temporali. La generica finestra di ricerca dell'applicazione web è riportata in Figura 25 (nell'esempio già compilato per la visualizzazione sincronizzata di 8 tipi di segnali differenti con campionamento a 10 minuti); oltre all'intervallo temporale, l'utente deve impostare il tipo di sensore (es. velocimetro), il sito (es. "Etna Cratere del Piano", ECPN), il canale di acquisizione (es. componente verticale HHZ), il tipo di serie temporale associata (es. RMS), il sotto-campionamento (es. 10 minuti). Una volta selezionati i parametri di ricerca per una data serie temporale, la richiesta viene aggiunta ad una lista (sulla destra del pannello I di Figura 25) cliccando sul tasto *Append Request*; a questo punto l'utente può inviare la richiesta al server (ricerca singola) oppure può continuare ad aggiungere richieste relative ad altre serie temporali, inoltrando tale richiesta multipla al server tramite il pulsante *Submit*. In Figura 26 è riportato un esempio di visualizzazione multipla di serie temporali; in particolare sono riportati i grafici del tremore sismico ed infrasonico relativi alle stazioni etnee di Cratere del Piano (ECPN) e Serra La Nave (ESLN), ed il grafico del radar ubicato al sito La Montagnola (i picchi sono relativi alle fontane di lava verificatesi nel periodo considerato).

Figura 25. Dettaglio del pannello di ricerca, con esempio di compilazione del *form*.

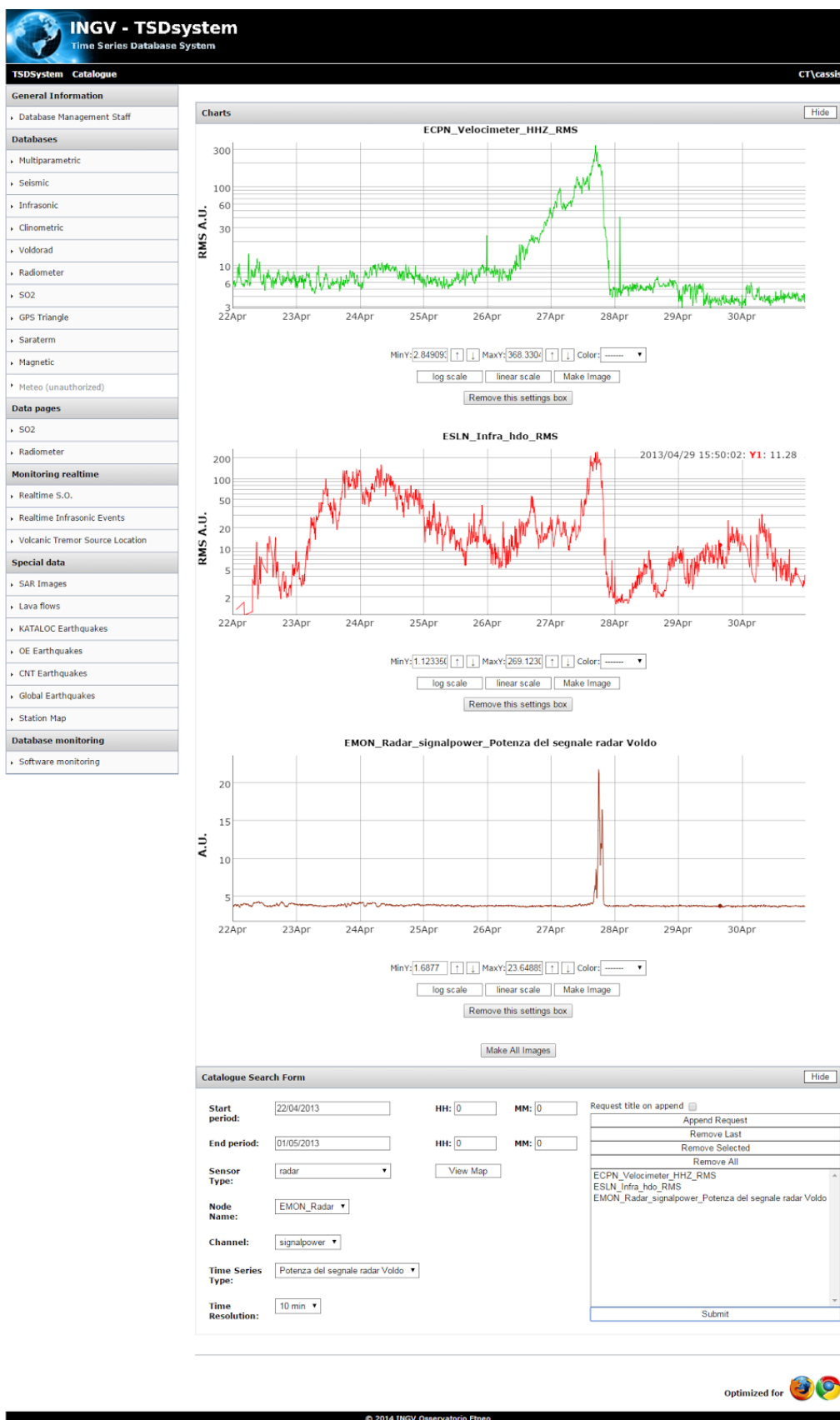


Figura 26. Esempio di visualizzazione multipla di serie temporali; in particolare sono riportati i grafici del tremore sismico ed infrasonico relativi alle stazioni etnee di Cratere del Piano (ECPN) e Serra La Nave (ESLN), ed il grafico del radar ubicato al sito La Montagnola (i picchi sono relativi alle fontane di lava verificatesi nel periodo considerato).

5. Conclusioni

L'approccio che si è dimostrato più promettente al fine di rilevare ed investigare le fasi di unrest vulcanico, nonché di prevedere le prossime eruzioni è il monitoraggio congiunto di dati vulcanologici, geofisici e geochimici [Scarpa e Gasparini, 1996; Tilling, 2008]. Negli ultimi anni, è stato ampiamente mostrato come l'analisi multiparametrico dei dati rappresenti la strategia vincente, non solo a scopi di monitoraggio ma anche per scopi di ricerca [Patanè et al., 2013].

L'applicazione sistematica di tale approccio non è un processo banale. In particolare, a causa dell'incremento del numero di sensori installati sui vulcani attivi, e della loro frequenza di campionamento, la quantità di dati che affluiscono continuamente all'interno degli osservatori sta crescendo in maniera esponenziale. Inoltre, le misure acquisite dalle differenti reti di sensori costituiscono sorgenti dati eterogenee. Al fine di gestire tutte queste problematiche, gli autori hanno sviluppato il framework *TSDSystem (Time Series and Spatial Database System)*, che oggi costituisce la Banca Dati di Sezione dell'Osservatorio Etneo dell'INGV, accessibile attualmente solo al personale della suddetta sezione. Una volta che sarà definita la politica di accesso ai dati, si provvederà a rendere il servizio fruibile a terzi per fini didattici/scientifici/di prevenzione.

Tale framework ci permette di acquisire flussi di dati da differenti reti di sensori localizzate sui vulcani siciliani standardizzandoli all'interno di un database relazionale.

A scopo riepilogativo, si riportano infine i servizi attualmente messi a disposizione dal framework TSDSystem:

- Gestione di stazioni multiparametriche.
- Sincronizzazione di serie temporali di dati multiparametrici.
- Visualizzazione di dati *real-time* e *near real-time*.
- Esecuzione di Query Avanzate utilizzando *web services* (mediante protocollo SOAP).
- Visualizzazione ed esportazione dei dati utilizzando formati web standard.

Ringraziamenti

Si ringrazia il personale tecnico dell'INGV-OE per la manutenzione delle reti di sensori. Parte dei risultati di questo report sono stati ottenuti nell'ambito del progetto PON01_00683 SIGMA (Sistema Integrato di sensori in ambiente cloud per la Gestione Multirischio Avanzata).

Bibliografia

- Aliotta, M., Montalto, P., Prestifilippo, M., Cannata, A., D'Agostino, M., Ferrari, F., and Torrisi, O. (2010). *Database to manage the instrumental networks run by Istituto Nazionale di Geofisica e Vulcanologia - Sezione di Catania: from planning stage to development* (in italian). Rapporti Tecnici INGV, 153.
- Cannata, A., Di Grazia, G., Aliotta, M., Cassisi, C., Montalto, P., and Patanè, D. (2013). *Monitoring seismo-volcanic and infrasonic signals at volcanoes: Mt. Etna case study*. Pure and Applied Geophysics, doi:10.1007/s00024-012-0634-x.
- Montalto, P., Aliotta, M., Cannata, A. and Cassisi, C. (2012). *Time series segmentation: an example of application on seismo-volcanic data* (in italian). Rapporti Tecnici INGV, 224.
- Patanè, D., Aiuppa, A., Aloisi, M., Behncke, B., Cannata, A., Coltelli, M., Di Grazia, G., Gambino, S., Gurrieri, S., Mattia, M., and Salerno, G. (2013). *Insights into magma and fluid transfer at Mount Etna by a multiparametric approach: A model of the events leading to the 2011 eruptive cycle*. J. Geophys. Res., doi: 10.1002/jgrb.50248.
- Scarpa, R., and Gasparini, P., *A review of volcano geophysics and volcano-monitoring methods*. In *Monitoring and mitigation of volcano hazards* (eds. Scarpa, Tilling) (Springer, 1996), pp. 3-22.
- Tilling, R.I. (2008), *The critical role of volcano monitoring in risk reduction*, Adv. Geosci. 14, 3-11.
- MySQL 5 Reference Manual (<http://dev.mysql.com/doc/#refman>).
- GNU GPL Documentation (<http://www.gnu.org/copyleft/gpl.html>).

Quaderni di Geofisica

ISSN 1590-2595

<http://istituto.ingv.it/l-ingv/produzione-scientifica/quaderni-di-geofisica/>

I Quaderni di Geofisica coprono tutti i campi disciplinari sviluppati all'interno dell'INGV, dando particolare risalto alla pubblicazione di dati, misure, osservazioni e loro elaborazioni anche preliminari, che per tipologia e dettaglio necessitano di una rapida diffusione nella comunità scientifica nazionale ed internazionale. La pubblicazione on-line fornisce accesso immediato a tutti i possibili utenti. L'Editorial Board multidisciplinare garantisce i requisiti di qualità per la pubblicazione dei contributi.

Rapporti tecnici INGV

ISSN 2039-7941

<http://istituto.ingv.it/l-ingv/produzione-scientifica/rapporti-tecnici-ingv/>

I Rapporti Tecnici INGV pubblicano contributi, sia in italiano che in inglese, di tipo tecnologico e di rilevante interesse tecnico-scientifico per gli ambiti disciplinari propri dell'INGV. La collana Rapporti Tecnici INGV pubblica esclusivamente on-line per garantire agli autori rapidità di diffusione e agli utenti accesso immediato ai dati pubblicati. L'Editorial Board multidisciplinare garantisce i requisiti di qualità per la pubblicazione dei contributi.

Miscellanea INGV

ISSN 2039-6651

<http://istituto.ingv.it/l-ingv/produzione-scientifica/miscellanea-ingv/>

La collana Miscellanea INGV nasce con l'intento di favorire la pubblicazione di contributi scientifici riguardanti le attività svolte dall'INGV (sismologia, vulcanologia, geologia, geomagnetismo, geochimica, aeronomia e innovazione tecnologica). In particolare, la collana Miscellanea INGV raccoglie reports di progetti scientifici, proceedings di convegni, manuali, monografie di rilevante interesse, raccolte di articoli ecc..

Coordinamento editoriale e impaginazione

Centro Editoriale Nazionale | INGV

Progetto grafico e redazionale

Daniela Riposati | Laboratorio Grafica e Immagini | INGV

© 2015 INGV Istituto Nazionale di Geofisica e Vulcanologia

Via di Vigna Murata, 605

00143 Roma

Tel. +39 06518601 Fax +39 065041181

<http://www.ingv.it>



Istituto Nazionale di Geofisica e Vulcanologia